Regis University
ePublications at Regis University

All Regis University Theses

Fall 2010

A Comparative Analysis of Auditing Within the Healthcare Database

Richard E. Butler
Regis University

A COMPARATIVE ANALYSIS OF AUDITING WITHIN THE HEALTHCARE DATABASE

A THESIS

SUBMITTED ON 25 SEPTEMBER, 2010

TO THE DEPARTMENT OF COMPUTER SCIENCE

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN

COMPUTER AND INFORMATION TECHNOLOGIES

BY

RICHARD E. BUTLER

APPROVALS

Glenn Balanoff, Thesis Advisor

Shari Plantz-Masters

Nancy Birkenheuer

Abstract

Healthcare database system utilization and processing demands are projected to increase significantly within the next decade due to governmental incentives, regulatory requirements, economic motivators and industry regulations. To that end, system engineers, architects, developers and solution providers will be pressured to meet these varied challenges while achieving or improving current system performance benchmarks. Healthcare database performance related to auditing has received little scholarly attention to date. A comparative analysis of two database auditing architectures is studied to identify efficiencies in auditing architectures within the healthcare domain. This study will explore and analyze native auditing supported by Microsoft's SQL Server 2008 Enterprise Edition and compare them to an EHR/EMR's implementation supporting the same auditing specifications. The result of this study will help to identify efficiencies in auditing architectures thereby assisting developers, architects and engineers in their efforts to implement efficient auditing, thereby improving overall system performance.

Acknowledgements

First of all, I would like to acknowledge my wife Lisa. She has always supported and encouraged me in everything that I have strived to accomplish. She is my rock. She is my best friend. I am truly blessed to have her by my side.

I would also like to thank my parents for instilling in me the moral character and work ethics that have served as my inner compass that has guided me through this wonderful life.

Lastly I would like to thank my thesis advisor, Mr. Glenn Balanoff for his guidance and advice in making this process enjoyable and enlightening.

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

Technology within the healthcare domain offers the ease-of-access to personal health information. Mobile devices and personal computers can easily provide user access to private health information (PHI) leveraging a variety of technologies. Private health information can thereby be accessed from virtually anywhere in the world utilizing any number of technologies. These technologies include but are certainly not limited to web portals, web services and even client\server applications over a virtual private network (VPN). The United States Congress identified the vulnerability of electronic private health information and first moved to protect PHI by enacting the Health Insurance Portability and Accountability Act (HIPAA) in 1996 (Hixson & Hunt-Unruh, 2008). Since the enactment of HIPAA, Congress has continued to enact similar legislation such as the Medical Privacy Rule (MPR) in 2003 and the HITECH act of 2009 which further regulate the protection of PHI and how it is shared (Wyne & Haider, 2007) (Stimulus Provisions Will Improve HIPAA, 2009) (Jingquan & Shaw, 2008).

From Paper Charts to Electronic Medical Records

A common misconception is that the regular record-keeping regarding patient encounters by physicians were adopted within the last two centuries. Conversely, the first evidence that physicians began to keep patient records was first noted in the writings of by Ali Al Rahawi who lived during the 9th century A.D. (Al Kawi, 1997). In his writings Rahawi explains that physicians are instructed that after entering a patient's place to visit, that they first call for a blank piece of paper on which they were to document the patient's signs, symptoms and what actions were taken by the physician. The physician would then leave the document with the patient. The first legislation associated with early physician notes was enacted by the Abbasid

Caliph in 931 A.D. (Al Kawi, 1997). The legislation not only required that documentation of patient visits be kept, but also contained the first documentation that legislated physician certification by examination. These humble beginnings of the patient chart empowered the patient to secure their medical records while providing a central location from which to obtain and review the patient's entire medical history; however, this practice was not widely adopted for centuries. Remarkably, recent legislation is striving to achieve this same level of centralization, security and access control of PHI over a thousand years later.

In the 1960's Dr. Larry Weed was first to present the concept of using an electronic system to store and collaborate patient medical information (nasbhc.org, 2000). Subsequently in 1972, The Regenstreif Institute developed the first electronic form of a medical record and decision making system known as The Regenstreif Medical Record System (McDonald, Murray, Jeris, Bhargava, Seeger, & Blevins, 1977). Although the Regenstreif Institute's success was revered as a technological breakthrough for medicine, it was not widely adopted by physicians (nasbhc.org, 2000). Subsequently in 1991, The Institute of Medicine released an influential report entitled "The Computer-Based Patient Record" which recommended all physicians implement EHR/EMR before the year 2000 (Woodward, 1995). The Institute of Medicine substantiated its recommendation by highlighting improved patient care and other benefits of utilizing EHR/EMR systems.

Despite the recommendations of well-respected organization such as The Institute of Medicine, widespread adoption of EHR/EMR systems have failed to be widely adopted. Why have healthcare providers, physicians primarily, not embraced EHR/EMR technologies? Research by David B. Meinert of Southwest Missouri University (Meinert, 2005) asserts that computer literacy, well publicized implementation failures, IT misconceptions and the overall lack of buy-in among physicians has managed to dampen the wide adoption of EHR/EMR systems historically. The overall maturity of information technologies, EHR/EMR systems, coupled with governmental incentives is projected to motivate significant adoption over the coming years (Meinert, 2005). Meinert also identifies that the increasing complexity of the practice of medicine has also contributed to a weak EHR/EMR adoption rate. As EHR/EMR systems mature with each new generation, so will the complexity, integration and system processing demands with each progression of each generation (Ball, Garets, & Handler, 2003)

Auditing the Healthcare Database

An integral part of HIPAA and subsequent federal regulations is the requirement that user activity and access of healthcare related information be audited (Jingquan & Shaw, 2008) (Wyne & Haider, 2007). The projected exponential adoption of Electronic Health Records/Electronic Medical Records (EHR/EMR) systems, due in part by recent healthcare reform (Ferris, 2008), will further increase database utilization and processing demands impacting global system performance (Ball, Garets, & Handler, 2003). As user and business requirements increase in complexity, so will the demands on overall system processing. It is due to this increase in access, utilization, user expectations and regulatory pressures that require the identification and implementation of efficient auditing architectures.

About this Study

A comparative analysis of two database auditing architectures is studied observing the simulation method. The results of this comparative analysis will provide valuable insight into efficient database auditing that will address current auditing requirements within the healthcare domain. This study will include and highlight the major aspects of auditing requirements within the healthcare domain. However, specific auditing requirements vary dependent upon various variables such as state, local, institutional and healthcare specialty (i.e., Ophthalmology, Audiology, Pediatrics, etc.) specific requirements. This study is focused on the general requirements and is neither all-encompassing nor prescriptive.

While this study focuses on Microsoft's SQL Server 2008, a paper study of Oracle's 11g R2 native auditing support will be conducted. The paper study will provide comparisons that identify differences and similarities in native auditing support between two major relational database management systems (RDMS). Other RDMS systems may also support native auditing functions and should be explored when implementing other platforms.

Chapter 2 - Auditing

Since the enactment of HIPAA in 1996, Healthcare providers and institutions have been required to audit authorized and un-authorized access to PHI. HIPAA addresses all types of PHI whether it is in paper or electronic form. Due to the physical and singular aspect of paper charts, implementing auditing functions regarding paper charts has been relatively simple. The emergence of copiers and faxing technologies has complicated access control and auditing of paper charts, but remains relatively simplistic. Complying with auditing regulations regarding electronic records presents unique and diverse challenges. Multiple users can access PHI from various physical locations concurrently. No longer can a single gatekeeper be responsible for authorizing and tracking access.

HIPAA requirements also include provisions that these audit logs be searchable and reportable. Further emphasizing auditing requirements and denoting them as non-voluntary standards, Table 2B – Adopted Privacy and Security Standards of the HITECH Act (Federal Register, 2010, p. 2035) states that “The date, time, patient identification (name or number), and user identification (name or number) must be recorded when electronic health information is created, modified, deleted, or printed. An indication of which action(s) occurred must also be recorded (e.g., modification).”

To date the healthcare industry has been slow to fully embrace technology (Wyne & Haider, 2007). The reluctance to adopt EHR/EMR systems has been attributed to the flexibility of entries as well as the support of narratives within paper charting (Kalra, 2006). A 2009 survey conducted by the Harvard School of Public Health found that only 1.5 percent of Acute Care Hospitals have a comprehensive EHR system while only 7.5 percent have a basic EHR and only 12 percent utilized electronic physician notes (EMR/EHR Utilization, 2009). An important aspect of the Health Information Technology for Economic and Clinical Health (HITECH) Act contained within the American Recovery and Reinvestment Act (ARRA) of 2009 significantly reduces Medicare and Medicaid reimbursements to healthcare providers that have not implemented EHR/EMR systems that meet “meaningful use” certification (Federal Register, 2010). The historical slow adoption of the healthcare industry to implement “meaningful use” EHR/EMR systems, coupled with significantly lower government reimbursement rates to healthcare providers that do not utilize “meaningful use” EHR/EMR systems, lends itself to a potentially enormous increase in utilization of EHR/EMR systems in the near term.

The projected growth and utilization of EHR/EMR systems will further stress system processing demands. In 2006 research that appeared in the Journal of the American Medical Informatics Association (JAMIA) projected that EHR/EMR adoption rates will increase from around 22% to over 86.6% by 2014 by the most conservative projects (Ford, Menachemi, & Phillips, Jan-Feb 2006) (*Figure 1* - Three Models of Projected EMR/EHR Adoption Rate). This study was conducted well before the HITECH act of 2009 which will have a direct influence on increasing EMR/EHR adopt rates.

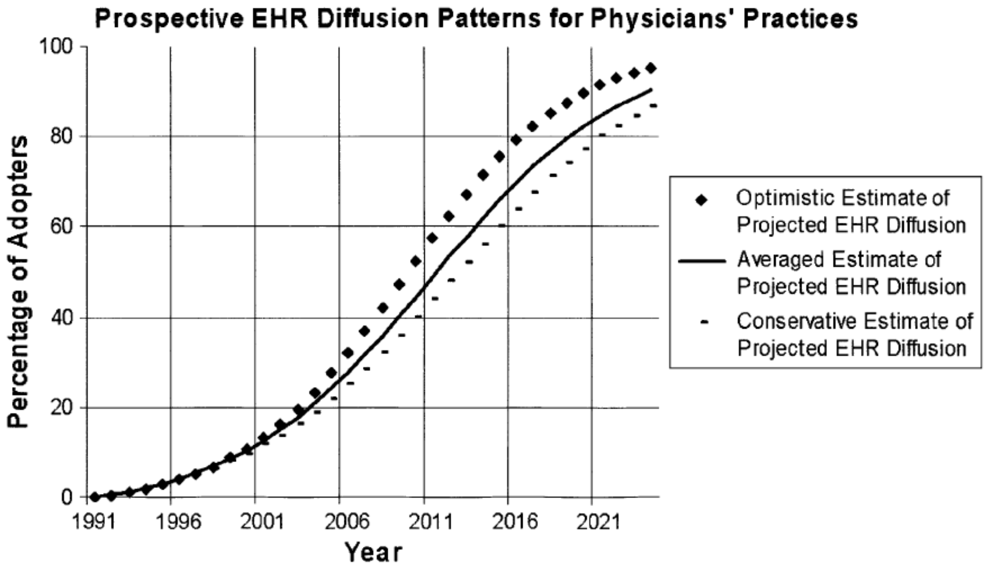

Figure 1: Three Models of Projected EMR/EHR Adoption Rate (Ford, Menachemi, & Phillips, Jan-Feb 2006)

Existing EHR/EMR systems will be required to support increasing concurrent users and transactions per second. Increased users and transactions will, by default, increase processing demands and further stress global system performance. EHR/EMR system engineers and architects will be tasked with providing more efficient processing that also satisfies compliance with auditing requirements.

Widespread adoption of EHR/EMRs is also gaining momentum from the growing complexity of medicine as well as increasing social pressures (Semerdjian, 2006). As the knowledge of medicine increases so does the complexity in collating and discerning more and more information more rapidly. Social pressures are realized as the general population and other industries embrace technology to perform other functions such as banking and other commercial industries. These functional expectations are naturally inferred to other industries such as access healthcare and sharing their medical records.

Chapter 3 – Current Auditing Positions

The most complete documents regarding auditing requirements within the healthcare database are in the form of government regulations. Several Department of Health and Human Services Code of Federal Regulations (CFR) specifically regulate that auditing occur, but falls short in addressing a specific architecture. HIPAA and the HITECH act are common abbreviations or terms that actually refer to one or more CFRs. For example, 45 CFR Parts 160 and 164 are commonly referred to as the HITECH act of 2009. These CFRs are the official regulations that outline auditing requirements that have to be met in order to avoid litigation, fines and ultimately lower reimbursement rates from governmental carriers if not satisfied by the specified timelines (Spicer, 2009). HIPAA, in its simplest terms, requires that entities that handle or have access to protected identifiable private health information are subject to regulation (Wyne & Haider, 2007).

Published in *The International Journal of Healthcare Information Systems and Informatics*, a 2007 study at the University of Michigan-Flint studied HIPAA compliant auditing within the healthcare database (Wyne & Haider, 2007). This study examined auditing implementing a JAVA JEEE solution that would serve as middle-ware applications residing between the application layer and the database layer. The purposed architecture reportedly would be easily and effectively implemented by auditing all applications that access the healthcare database by utilizing middleware components. The study identifies that historically, healthcare institutions have implemented information systems in an ad-hoc manner which lacked a long term plan to ultimately provide an enterprise wide EMR/EHR solution. By implementing a middleware application that audits access and user activity, legacy systems could be easily retrofitted to become compliant without the need to re-factor and redeploy legacy systems. While the Michigan-Flint approach provides a solution in the near term, the goal of the current study strives to examine a logical next evolution in the auditing architecture of healthcare database systems.

Vendor Auditing

EHR/EMR vendor specific auditing architectures will naturally vary from vendor to vendor as will their efficiency. To study all of them would be never-ending since new EHR/EMR systems enter the market frequently. Due to the time constraints related to this study, a single vendor will be identified and selected but will remain anonymous. EHR/EMR vendors are encouraged to study and analyze their respective auditing architectures and compare them to this study to assist them in optimizing their respective auditing architectures.

The EHR\EMR system that was selected has been in the healthcare market for approximately twelve years and is comprised of several domain specific products that provide ambulatory surgical care, routine office exams and imaging modules. Customers can opt to purchase one module or the entire suite dependent upon specific practice needs. The price of each module are independent but are discounted when purchasing the entire suite. Pricing is not publically published and is calculated on a client-by-client and specific needs of a given client. The selected system also supports a number of interfaces common within the healthcare domain. An example of this is would be the support for Healthcare Level 7 (HL7) interfaces as a means to interface and exchange specific patient and exam information with external systems. These external systems are typically enterprise document repositories or other EMR\EHR systems such as IC Chart, Epic, Practice Partners, Allscripts or other commercial healthcare products that also provide support for HL7 communication. According to the company’s website, the selected system has an install based of over three hundred clinics processing over an estimated fifty thousand patient exam visits per day. During normal operations at larger client sites, it is estimated that there are typically an estimated 200 concurrent users during normal clinic hours.

The selected system is a Windows based application that is designed and optimized specifically for SQL Server. According to the company web site, the system can be installed in a number of variations which include workstation fat-client, terminal server, Citrix or thin terminal configurations. Currently the system is not natively compatible with non-windows based operating systems or non SQL Server database platforms. The most current release documentation states that the system is compatible with x86 and x64 based windows operating systems and is supported on Windows XP, Vista and Windows 7. The product documentation also states that x86 and x64 versions of SQL Server 2005 Standard Edition and higher are supported. However, previous versions of SQL Server and Express or lower editions than Standard edition are not supported. Enterprise edition of SQL Server is required in certain configurations.

Based on information obtained from the company's web site, during the implementation phase of the product, a certain number of client users are identified to full-fill the role of "Super User." The role of the Super User is to become an internal expert and leader in regards to the functionality and management of the system. Super Users are charged with training other users as needed as well as an internal subject matter expert. To provide user support needs, customers have access to continuing education through on-line or onsite training. During each release cycle, the Super Users are given training and documentation on product changes and enhancements pertaining to the new versions. If deemed necessary, remedial and continuing user training and support are provided through on-site and online training are available. Clients also have access to product user groups. An annual user's group meeting is held at the corporate headquarters to provide a platform for communication and education regarding each product.

Technical support is also provided and included within maintenance contracts. Routine technical support is provided daily from 8:00 a.m. to 10:00 p.m. eastern standard time with emergency access when needed by contacting a published technical support phone number. Technical support can also be provided on-site as needed for specific circumstances when it is required, but is usually provided through phone as well as remote access to the client's system using virtual private network (VPN) access and remote desktop technologies.

The EHR/EMR system that was selected observes the traditional client\server application architecture which implements a windows client application accessing a centralized SQL Server database. Modern EHR/EMR systems architectures vary from traditional client/server to web-based to hybrid offerings that marry web and win forms technologies (Nash & Goldfarb, 2006).

The selected system performs auditing of user actions from within the application. Auditable user actions are detected and audited from within the application itself (*Figure 2* - View Chart Audit Sequence Diagram). An example of an audited event would be a user selects to view a patient's medical exam chart from a previous visit. The user event of selecting the patient's exam for viewing results in the firing of an audit condition. The event in turn calls the auditing process to insert the pertinent information regarding the event into the audit table. An audit stored procedure is called to insert the data into the audit table.

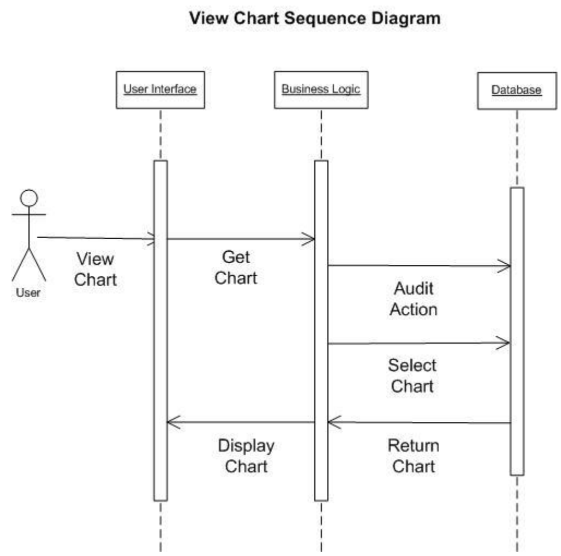

Figure 2 – View Chart Audit Sequence Diagram

The vendor auditing architecture uses a SQL Server table to hold audited actions. An auditable user action within the EHR/EMR application triggers to perform an insert into the audit table. The audit table contains the patient medical record number, date time stamp, user login, object accessed and DML actions (Select, insert, update, delete) that were performed.

SQL Server 2008 Auditing

Introduced with Microsoft's SQL Server 2008, native auditing was engineered to provide an integrated and highly efficient means to audit database and server object activity. This was facilitated by the introduction of a high-performance eventing infrastructure called SQL Server Extended Events (Lee & Rask, 2009). Auditing in previous version of SQL Server were supported by implementing custom SQL Traces which were not engineered to specifically address auditing and were found to be lacking in functionality and efficiency (Lee & Rask, 2009). As of this study, Microsoft's SQL Server 2008 Enterprise Edition is the latest release that provides support for native auditing of various server and database level object events. Auditing events can be customized and defined to capture specific events by specific users or groups of users at the object. Objects reside at the server or database level and thereby are managed accordingly. SQL Server auditing is comprised of three main components. These components are defined as Server Audit, Server Audit Specification and Database Audit Specification. An overview of the architecture and relationships between these objects is represented in *Figure 3* – SQL Server Audit Object Layout.

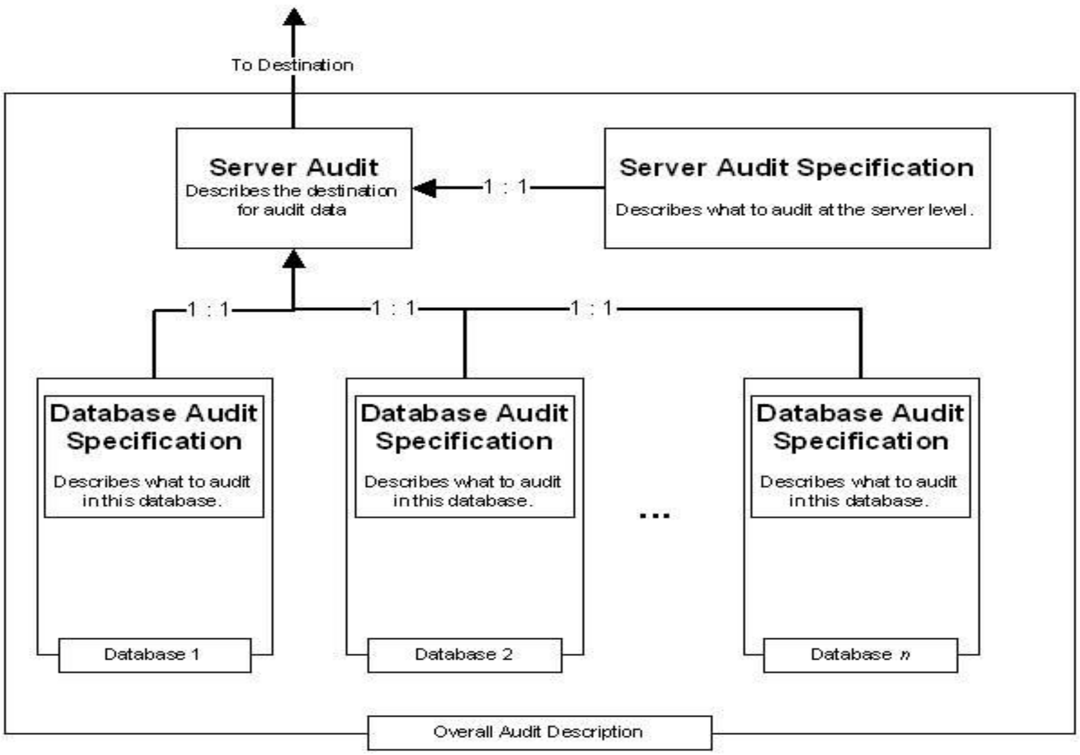

Figure 3: SQL Server Audit Object Layout (Lee & Rask, 2009)

Creating and Configuring the Server Audit

The Server Audit object defines the target or destination for the audit data. Audit data can be logged to one of three destinations. Audit data can be logged to the windows application log, the windows security log or to an encoded file written to disk. A Server Audit object can be created using SQL Server Management Studio (SSMS) or by executing T-SQL statements. For the purpose of demonstration, this paper will illustrate the T-SQL method.

To create a server audit, the CREATE SERVER AUDIT T-SQL statement, altered using the ALTER SERVER AUDIT or removed using the DROP SERVER AUDIT T-SQL statements. When creating a server audit one of the most important factors, besides giving the audit an intuitive and descriptive name, is to plan how the audit data will be queried. There are a number of third party tools that will query various windows logs and are very useful in searching windows logs. If audit data is written to a file on the other hand, it can be queried using T-SQL from within any T-SQL query application like SQL Server Management Studio. Writing audit data to a file yields a very flexible and powerful tool for querying large amounts of audit data. For the purpose of this study, all data will be written to a file located within the C:\SQL\Audit directory that has been created. When writing audit data to a file, several options are available and should be considered (*Table 1 - Optional SQL Server Audit Specifications*).

Table 1 – Optional SQL Server Audit Specifications

Option	**Description**
TO	Available values are FILE, APPLICATION_LOG or SECURITY_LOG. If FILE is defined, it must be followed by the full path to the folder where the audit files are to be written.
MAXSIZE	This option sets the maximum size of the audit file. When this threshold is met, a new log file will be created. All subsequent audit data will be written to the new log file.
MAX_ROLLOVER_FILES	This value represents the maximum number of files that will be created. When this value is reached, SQL Server will then begin to delete the oldest log file when a new log file is created.
RESERVE_DISK_SPACE	Value can be “ON” or “OFF”. Is on, SQL Server will reserve the

	defined disk space at when SQL Server Services start after a shutdown or reboot. The reserved disk space can then be allocated and used by auditing.
QUEUE_DELAY	By default audit data is written synchronously with a QUEUE_DELAY of 0. If this value is greater than 0, audit data will be written asynchronously.
ON_FAILURE	Values are “SHUTDOWN” or “CONTINUE”. If for any reason the writing of the audit data fails, the defined action will take place. SHUTDOWN will stop processing and the user action will not occur.

To comply with the requirements of this study, SQL Server auditing was created and enabled to capture successful as well as failed logins to the SQL Server. For simplicity, only the tables identified will be audited during this study. Additional tables can be identified and added to the audit criteria using the Alter Audit Specification including the ADD argument to add additional database objects as needed. Refer to Appendix A for the T-SQL statements used to create the various audit objects for this study.

Chapter 4 – Research Methodology

Test Environment

This study examines two auditing architectures and strives to determine which architecture is more efficient. To accomplish this goal a testing environment is required to provide comparable and reliable results. To form the basis of the test environment, a personal computer system was purchased. Purchasing a personal computer provides a sound and current hardware base upon which to install current software. Once acquired, a current server class operating system and related software packages, including service packs, were installed further ensuring that the test environment is current and relative to modern real-world implementations. A local area network (LAN) was established to provide a realistic test environment in which to conduct the performance tests using two laptop computers. One laptop computer will be dedicated to host Spotlight on SQL Server™ (SOSS) by Quest Software (Appendix C). SOSS is a commercially available enterprise class product used to assist Database Administrators in monitoring SQL Server performance within an enterprise environment. A second laptop computer is dedicated to host custom developed multi-threaded software that is used to stress the test environment. The stress testing software will be used to simulate 30 concurrent users performing a routine exam. Built-in thread execution offsets are implemented to simulate real world concurrent users. *Table 2 – Test Environment Hardware\Software Specifications* illustrates the specific hardware and software configurations of the test environment.

Table 3 – Test Environment Hardware\Software Specifications

Operating System	Windows Server 2008 R2 x64 Enterprise Edition
SQL Server	SQL Server 2008 x64 Enterprise Edition
CPU	AMD Athlon x64 1.60GHz

Memory	2.00 GB
Disk Drives	Single 160 GB Drive

Test Environment Configurations

HIPAA does not provide specific details regarding auditing requirements, but rather provides a broad set of requirements. It is not the intent of this study to interpret HIPAA or other auditing requirements absolutely or definitively, but rather to examine performance indicators associated with two auditing architectures implementing the same level of auditing granularity related to the same database objects under identical workloads. The identified objects are believed to be core components to satisfying these requirements and were identified during the normal process of completing a routine patient exam using the selected commercially available EHR/EMR product. SQL Server Profiler was used to capture a trace file of SQL objects that were accessed during the simulated test examination. The results of the SQL Server Profiler trace were then used as the basis to identify database objects accessed during the course of the exam. The SQL Server Profiler trace file was then used to develop two workload scripts that were executed against each test environment.

To provide a realistic EHR/EMR model database, a SQL Server database was created based on an obfuscated private practice database using the EHR/EMR system. All PHI information contained within the database was obfuscated using T-SQL to update the patient identifying information. A backup was then taken of the obfuscated database and restored to the test environment to ensure that the test environment provided a mirror image of a real world implementation that contained realistic patient data. In reviewing the SQL Server Profiler trace file, it was noted that the vendor had implemented an audit table named AUDIT_LOG to hold auditing related information. T-SQL statements were used to insert the related action (selects, inserts, updates or deletes) into the AUDIT_LOG table whenever a user selected, inserted, updated or deleted auditable data within the database. Additionally, the AUDIT_LOG table captured values for related date, time, table name, user name and host name. The vendor audit workload will include these T-SQL statements; however, these audit T-SQL statements were omitted from the SQL audit workload since they were replaced with native auditing. Great care was taken to ensure that the SQL Server audits captured the same degree of granularity as the vendor T-SQL audits.

SOSS was installed and configured to monitor the test environment. SOSS was configured to monitor operating system, network and SQL Server performance indicators. Stress testing software was developed in Visual Basic .Net to simulate 30 concurrent users executing the T-SQL commands. These T-SQL commands were capture and extracted from the SQL Server Profiler trace file that was captured during the performance of a routine exam. The only exception to this is the omission of the T-SQL commands that executed inserts into the vendor's AUDIT_LOG table. The AUDIT_LOG T-SQL commands were replaced with SQL Server audit specifications that mimicked the same degree of logging.

Each thread execution within the workload scripts were offset by 300 milliseconds and each T-SQL command was executed every 1000 milliseconds to simulate as close to real world concurrent user activity as possible. The stress testing software was developed to write a timestamp, the executed T-SQL statement and which thread executed the statement. A total of ten executions of the stress test script were executed against each environment giving the ability to average the results. The average of the results will provide a more accurate interpretation of performance. *Figure 4* – Testing Lab Configuration illustrates the test environment configurations and how they will be stressed and monitored during this study.

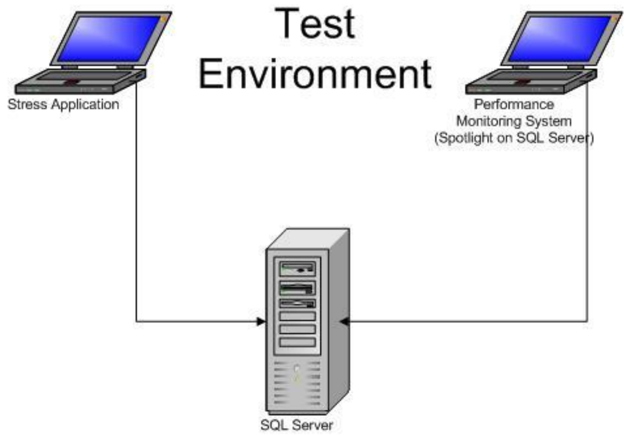

Figure 4 - Testing Lab Logical Configuration

Capturing and Measuring Performance Indicators

This study implements the simulation model of performance evaluation. Simulation modeling is more accurate than other database performance models because it not only analyzes the database system but also analyzes the database system under realistic application data against it (Paul & Jain, 2008). To accurately gauge effects of any process or action, two variables must be known. These variables are the baseline (at rest) value and the stressed value when a workload is applied (Paul & Jain, 2008). Additionally, SQL Server 2008 consists of a number of automated backend processes that are essential to SQL Server's operation (Chaudhuri, Christensen, Graefe, Narasayya, & Zwilling, Bulletin of the Technical Committe on Data Engineering, 1999). These processes include but are not limited to SQL Server Query Optimizer, Buffer Manager, Lazy Writer, Resource Monitor and Lock Manager. These backend processes can be administered and configured to a degree, but are largely black box processes that execute when deemed necessary by SQL Server services. Naturally when these processes execute, they affect performance indicators that are targeted by this study. To that end, it is imperative to minimize this extraneous data from computations related to this study. In an effort to mitigate extraneous influences to performance indicators such as these backend processes, a baseline as well as stress results will be averaged at predefined intervals during the testing period. Additionally, between each test cycle, SQL Server and related services will be stopped and restarted to release any allocated resources such as memory.

Average indicators were obtained and calculated by capturing the indicators every 15 seconds for a 10 minute time period. The equation $a = \frac{i}{40}$ was used to calculate the average baseline value for each performance indicator. The variable *i* represent the sum of all values captured for a given performance indicator at each interval. This average baseline value will be compared to the average value of the same indicator while under stress yielding the true effects of a specific auditing architecture relative to a specific indicator during the stressing periods.

The effects of test workloads were measured in terms of denoting average values for each performance indicator during the stress period. Performance indicator values were captured every 15 seconds for the duration of the stressing period. The equation $a = \frac{i}{n}$ was used to calculate the average stressed value for each performance indicator. The variable *i* represents the sum performance indicator values while *n* represents the total number of values captured.

To accurately and recursively evaluate performance indicators, a video recording of SOSS was capture using Camtasia® video screen capturing software from Techsmith. Video capturing of the baseline as well as during the stressing period enables this study to review and scrutinize performance indicators in depth and at will.

Chapter 5 – Results of the Research

A baseline had to be established to provide a foundation for a comparative analysis between an at-rest and an at-stressed state. To establish the at-rest baseline, the seventeen performance indicator values were observed and catalogued into an Excel spreadsheet every fifteen seconds during an at-rest period of 10 minutes. The results yielded a total of forty discrete values for each of the performance indicators. The average baseline for each performance indicator was then calculated using the formula $a = \frac{i}{40}$.

The test workload scripts were executed a total of ten consecutive times against each audit configuration. Each execution simulated thirty concurrent users against the audit configuration with a 300 millisecond offset between each user's simulated start times. At the end of each stressing period, SQL Server services were stopped and restarted to release allocated resources during the stressing period. This recycling of SQL Server ensures that baseline values remain equitable between stressing periods.

Stressed values were observed and catalogued into an Excel spreadsheet every fifteen seconds during the stressing period yielding a total of 76 value sets for each environment (Appendix B). The formula $a = \frac{i}{76}$ was then used to calculate the averaged stressed value for each of the seventeen performance indicators. All values were rounded to three decimal places for comparison. *Table 3 – Averaged Performance Indicator Values* illustrates results of the calculated baseline and performance indicator values. The seventeen performance indicators relative to each of the two auditing environments and their respective averaged values are noted.

Table 4 – Averaged Performance Indicator Values

Matrix	Baseline	SQL Audit	Vendor Audit

% CPU	7.25	13.566	13.816
Total Memory in MB	79.550	128.270	135.928
Buffer Cache in MB	24.143	43.341	45.514
Procedure Cache in MB	46.955	76.854	82.338
Procedure Cache Hit Ratio	77.19	91.818	92.784
Disk I/O wait time in millisecond	0.980	6.427	4.790
Logical Reads per second	125.725	615.342	853.961
Compiles per second	4.298	1.386	1.468
Packets sent per second	1.070	26.907	28.957
Packets received per second	1.308	26.547	27.312
Batches per second	1.055	26.269	26.957
Physical Reads per second	0.013	2.366	2.327
Physical Writes per second	0.033	0.031	0.236
Read ahead pages per second	0	0.497	0.389
Checkpoint Pages per second	0	0	0
Lazy Write Pages per second	0	0	0
Log Flushes per second	0.104	0.367	25.038

The results of the tests were cataloged and analyzed to determine which performance indicators were affected and how significant the impact was when compared to the averaged environmental baselines. Of the seventeen performance indicators that were measured, indicator values within the vendor auditing environment increased more than within the SQL Server audit environment during the stressing period. The SQL audit environment did have increases that outpaced the vendor audit environment in regards to disk I/O wait, time physical reads and read ahead pages per second. Of the seventeen performance indicators, twelve out of sixteen were noted to have significantly increased within the vendor auditing environment while two indicators, Checkpoint pages per second and Lazy write pages per second, experienced no changes from the baseline values.

Chapter 6 – Analysis of Results

The performance indicators captured during the research were analyzed to identify the overall impact the workload had regarding each of the various performance indicators. Performance indicators were analyzed individually and then in logical groups to accurately identify how each auditing environment performed during the stressing period when compared to baseline values for the same performance indicators. The overall evaluation of the results will provide insight into determining the overall impact the work load had when executed against each environment.

Individual Performance Indicator Analysis

One of the most prevalent performance indicators is percent CPU utilization (Nichter, 2007). SQL Server is tightly integrated into the Windows operating systems which allows for SQL Server percent CPU utilization to be isolated and captured within SOSS and other performance monitoring software such as Performance Monitor. Capturing the percent SQL Server CPU performance indicator specifically excludes non-SQL Server CPU processes thereby improving the accuracy of the percent CPU utilization values in regards to SQL Server performance (Fritchey & Dam, 2009). CPU Utilization was calculated at 13.816% in the vendor auditing environment while the SQL Server audit environment realized only 13.566% (*Figure 5* – Percent CPU Utilization). However, an increase in CPU utilization alone is not an indicator of inefficiency. SQL Server is architected to dynamically utilize available CPU processing capabilities when needed and can be more than two times as efficient as even the latest DDR3 RAM (Nichter, 2007) (Fritchey & Dam, 2009).

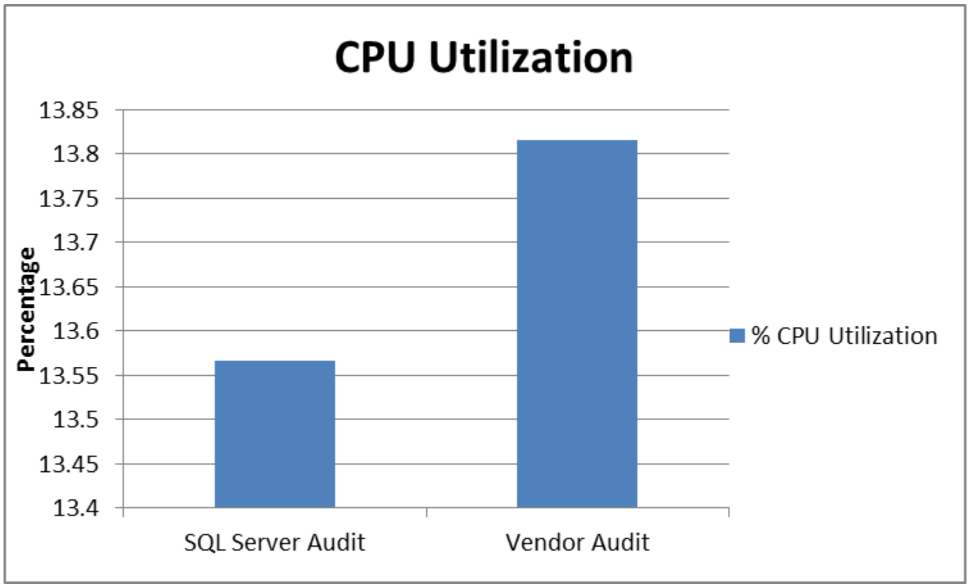

Figure 5 – Percent CPU Utilization

Memory utilization with SQL Server is allocated dynamically by default. SQL Server will dynamically allocate additional memory to strive to improve performance, mainly related to procedure and buffer cache hit ratios (Fritchey & Dam, 2009). During the vendor audit stress period 135.928mb of memory was allocated compared to 128.27mb during the SQL Server audit stressing period. While this is significant, SQL Server is architected to dynamically allocate memory as needed by SQL Server processes based on available memory (Chaudhuri, Christensen, Graefe, Narasayya, & Zwilling, Bulletin of the Technical Committe on Data Engineering, 1999) (Fritchey & Dam, 2009). Additionally, SQL Server will only release allocated memory if the memory is subsequently needed by the operating system for other processes. If the operating system does not require additional memory for other processes, the memory will remain allocated to SQL Server (Chaudhuri, Christensen, Graefe, Narasayya, & Zwilling, Bulletin of the Technical Committe on Data Engineering, 1999). This can also be realized by the fact that once memory was acquired by SQL Server during the on-set of the stressing periods, the allocated memory remained constant throughout the stress period and persisted after the stressing periods had ended. (*Figure 6* – Memory Utilization). It was noted that during the vendor audit stress period, an additional 7.658mb of memory was allocated than what was allocated during the SQL Server audit stressing period.

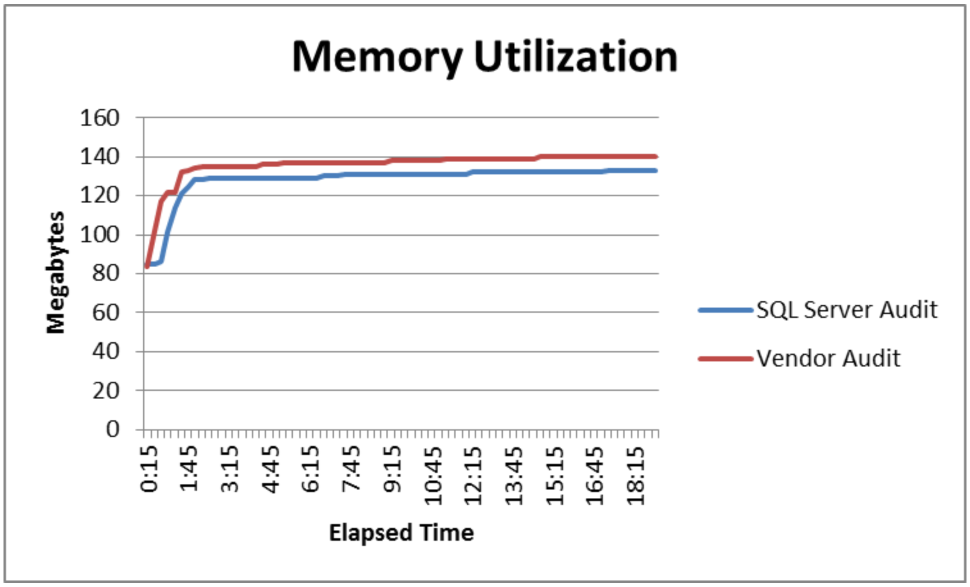

Figure 6 - Memory Utilization

Memory utilization in SQL Server can be configured to utilize specific minimum and maximum values (configured values) effectively by-passing dynamic (running values) memory allocation. Dynamic memory allocation is enabled by default; however enabling running values is typically more suitable for servers that host other applications such as Microsoft Exchange Server (Fritchey & Dam, 2009). The presence of two memory intensive applications on the same physical server will commonly result in memory contention between the two applications. In these circumstances, it is recommended to enable configured memory allocation ensuring that SQL Server is allocated sufficient memory and thereby eliminating memory contention (Fritchey & Dam, 2009).

The ability for SQL Server to read and write to hard disk is pivotal to overall system operations. Latency in disk I/O directly results in delays in SQL Server’s ability to read and write data pages which negatively impacts overall system performance. Disk I/O bottlenecks are a common cause of poor performance in SQL Server (Fritchey & Dam, 2009). Therefore, an increase in disk I/O demands is a main performance indicator of concern in regards to global system performance. However, it is worth noting that disk I/O latency may be due to excessive paging which is usually due to insufficient memory (Fritchey & Dam, 2009).

Maximum hard disk drive I/O wait times were noted to increase more during the SQL Server audit stress period when compared to the vendor audit stress period (*Figure 7* – Maximum Disk I/O Wait Time). During the stressing periods, both the SQL Server and the vendor audit tests were noted to experience a number of significant short spikes in maximum disk I/O wait times. These spikes were noted to be significantly higher than 25 milliseconds during both stressing periods. It is worth noting that the test environment hardware configuration consisted on a single 160 gigabyte physical SATA II 7,200 rpm disk drive which is not consistent with recommended hardware requirements from Microsoft. The recommended hardware configuration would typically consist of a dedicated hard disk drive or storage area network (SAN) logical unit number (LUN) for each data and log file (Ruthruff, 2007). Dedicating a physical hard drive or LUN will provide sufficient disk I/O through-put to support parallel disk reads and writes exponentially increasing disk I/O capacity.

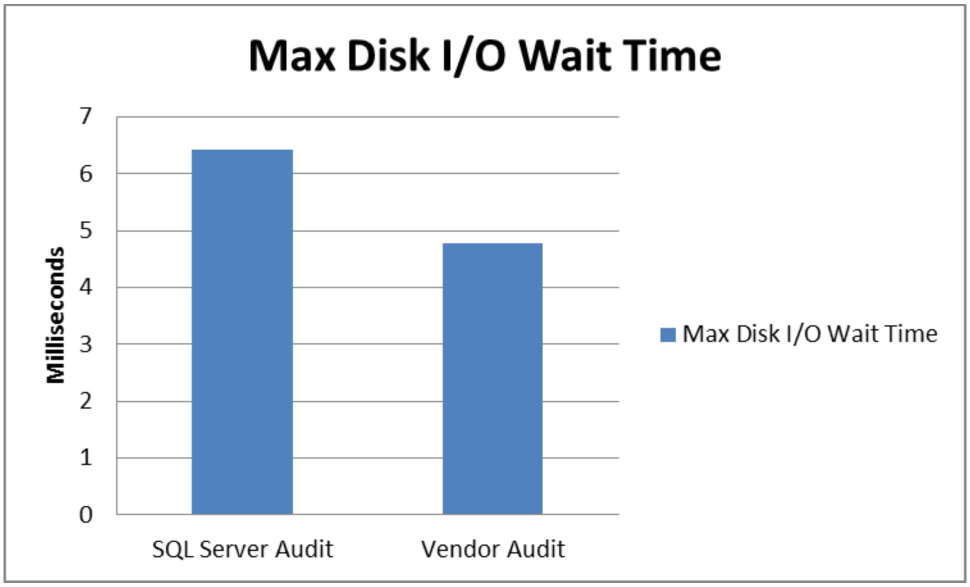

Figure 7 – Maximum Disk I/O Wait Time

Persistent disk I/O latency of more than 10 milliseconds can indicates a disk I/O bottleneck and can significantly degrade SQL Server performance (Fritchey & Dam, 2009).

Logical reads are a performance indicator that represents the read operations that were performed by queries. The higher the number of logical reads per second, the higher the disk read operations. Logical reads reflect that the page or pages requested by the query has to be read from disk into memory (Fritchey & Dam, 2009). Cursors can be a major factor in elevating logical reads per second; each iteration of a cursor requires additional logical read operations be executed.

During the stressing period, logical reads per second were also noted to have increased significantly during the vendor audit stressing period when compared to the SQL Server audit stressing period. Logical reads per second during the vendor audit stressing period averaged 853.961 per second while during the SQL Server audit stressing period logical reads were noted to average only 615.342 per second (*Figure 8* – Logical Reads/second).

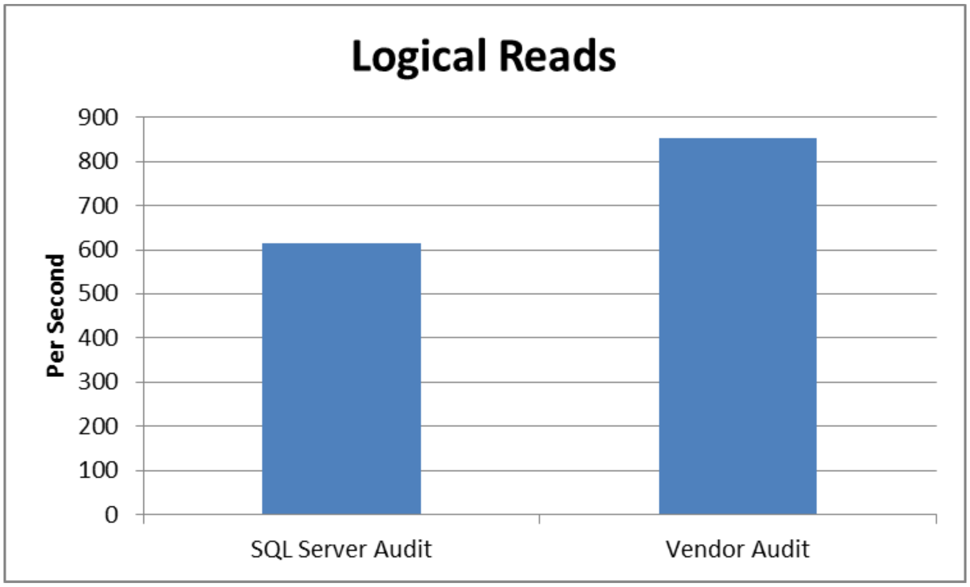

Figure 8 - Logical Reads per second

To increase overall system performance, SQL Server caches compiled query plans for queries and are most efficient when utilized within stored procedures. When SQL Server compiles a query, the resulting query plan should reflect the most efficient query plan based on current indexes, statistics and data contained within the underlying tables. The compiled query plan can then be utilized should the query be re-executed while the query plan remains in the procedure cache. Additionally, a certain number of recompiles are expected and healthy. However, recompiles are CPU intensive and an inordinate number of recompiles, commonly due to insufficient memory, can result in excessive CPU utilization (Fritchey & Dam, 2009).

SQL Server compiles per second were noted to increase approximately 44% during the vendor audit stressing period over the SQL audit stressing period. The vendor audit stressing period experienced an average of 1.468 compiles per second whereas during the SQL Server audit period realized only an average of 1.386 compiles per second (*Figure 9* – SQL Server Compiles per second).

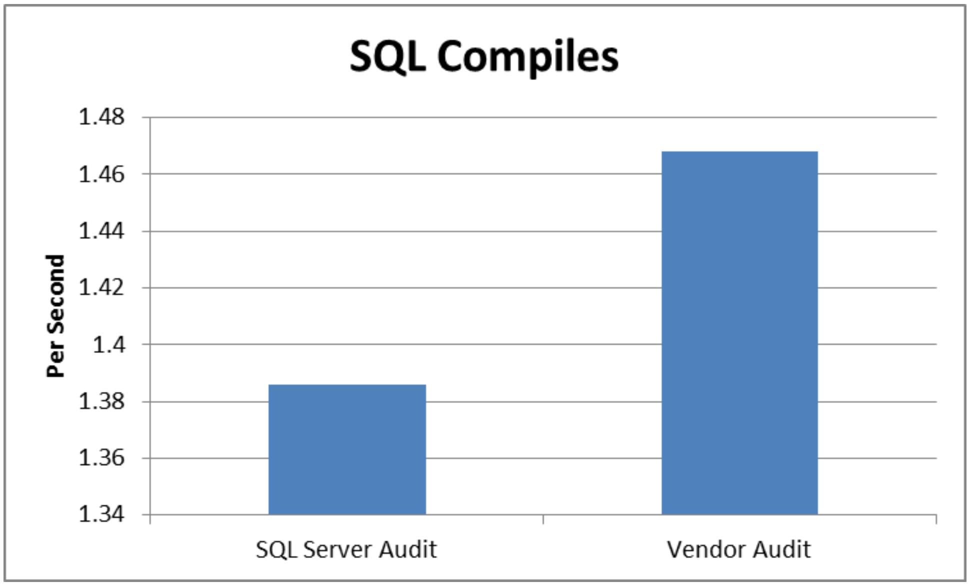

Figure 9 – SQL Server Compiles per second

To increase performance SQL Server implements a buffer cache which is used to hold recently retrieve data. The goal of the buffer cache is to store frequently or recently queried data in cache to eliminate the slow and more costly process of retrieving the data from much slower disk drive read operations.

Buffer cache increased from an average of 45.514mb during the vendor auditing stressing period. Conversely, during the SQL Server audit stressing period realized an average of only 43.341mb (*Figure 10* – SQL Server Buffer Cache).

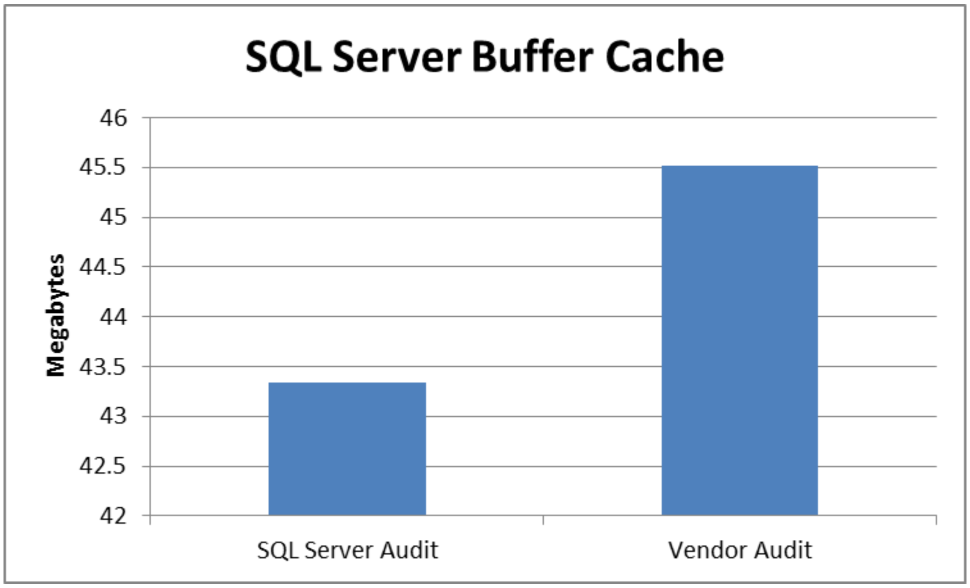

Figure 10 - Buffer Cache

To optimized performance SQL Server also implements a procedure cache. The procedure cache contains compiled query plans for recently executed SQL queries. By storing these pre-compiled query plans in a cache, SQL Server can utilize these optimal query plans and thereby eliminating the less efficient process of determining (compiling) a new optimal query plan. The vendor audit stress period realized a gain of 82.338mb compared to an increase of 76.854mb during the SQL Server audit stressing period. (*Figure 11* – SQL Server Procedure Cache).

A significant component in analyzing procedure cache is the hit ratio. The procedure cache hit ratio reflects what percentages of query plans are located within the procedure cache. A query plan that is located within the procedure cache is far more efficient since a recent and relevant query plan can be executed and does not require the costly process of compiling a new query plan. During the vendor audit stressing period, it was noted to have a slightly higher procedure cache hit ratio (92.784%) than did the SQL Server audit test (91.818%).

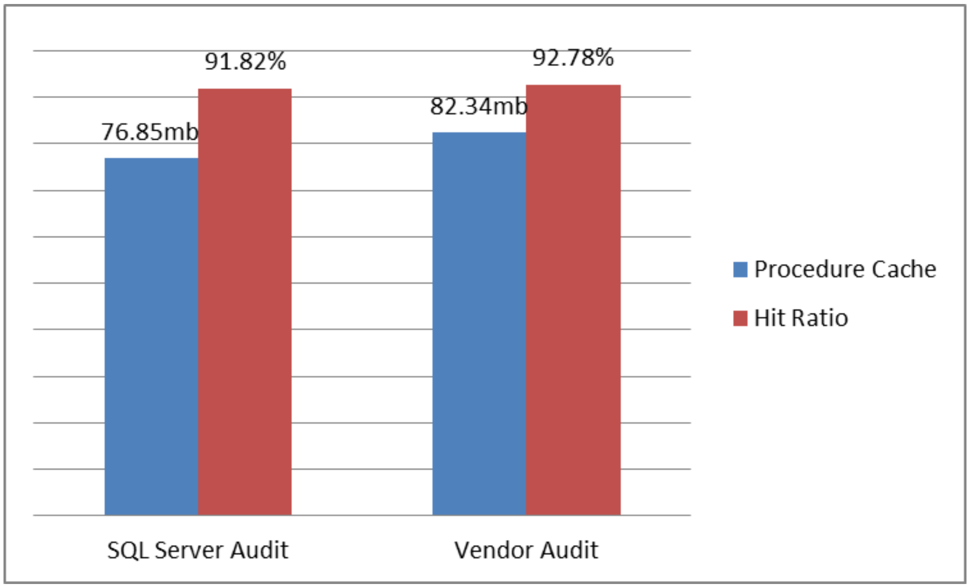

Figure 11 – SQL Server Procedure Cache

Client server application such as the focus of this study, as well web based applications, utilize network communication protocols such as open database connectivity (ODBC) technologies to send and receive packets to a centralized RDMS such as SQL Server (Lewandowski, 1998). Excessive network communication or chatter between the clients and the RDMS will not only affect a specific application, but overall network performance for unrelated applications, such as print tasks, will also be affected (Fritchey & Dam, 2009). Therefore, network bandwidth utilization in terms of packets per second sent and received was monitored during this study.

The SQL Server audit stressing period was observed to send and receive fewer packets per second when compared to the vendor audit stressing period. The vendor audit stressing period averaged 28.957 sent and 27.312 received while the SQL Server audit stressing period averaged 26.907 and 26.547 respectively (*Figure 12* – Packets Sent\Received per second).

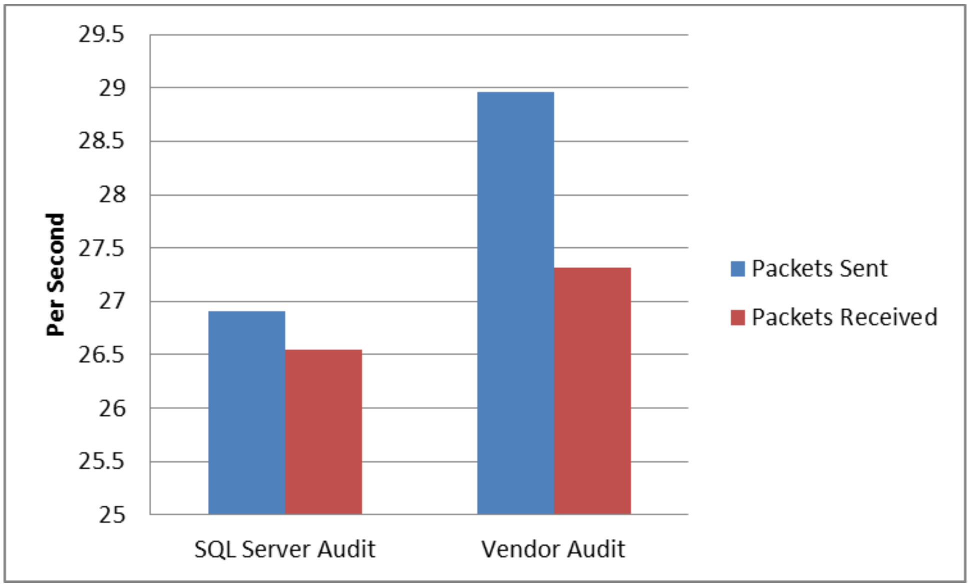

Figure 12 - Packets Sent\Received per Second

To respond to request for data, as well as perform updates to existing data, SQL Server performs physical read and write operations. Physical read operations require disk I/O operations to read the required page or pages from the physical disk if the requested data page or pages are not already present in the buffer cache. Updates or inserts of new data pages requires that SQL Server write these updates or add new data pages to the physical log file. Once the log file is committed and subsequently flushed, SQL Server will then write the related data pages to the physical data file within the defined file group (Fritchey & Dam, 2009).

Physical reads and writes were more elevated during the vendor audit stressing period when compared to the SQL Server stressing period (*Figure 13* – Physical Reads\Write per Second). The SQL Server audit stressing period realized a 2.366 reads per second while the vendor audit stressing period realized a slightly higher average increase of 2.327 reads per second. Physical writes per second during the SQL Server audit stressing were observed to increase to 0.031 writes per second while the vendor audit stressing period was observed to experience a slightly higher increase 0.236 writes per second.

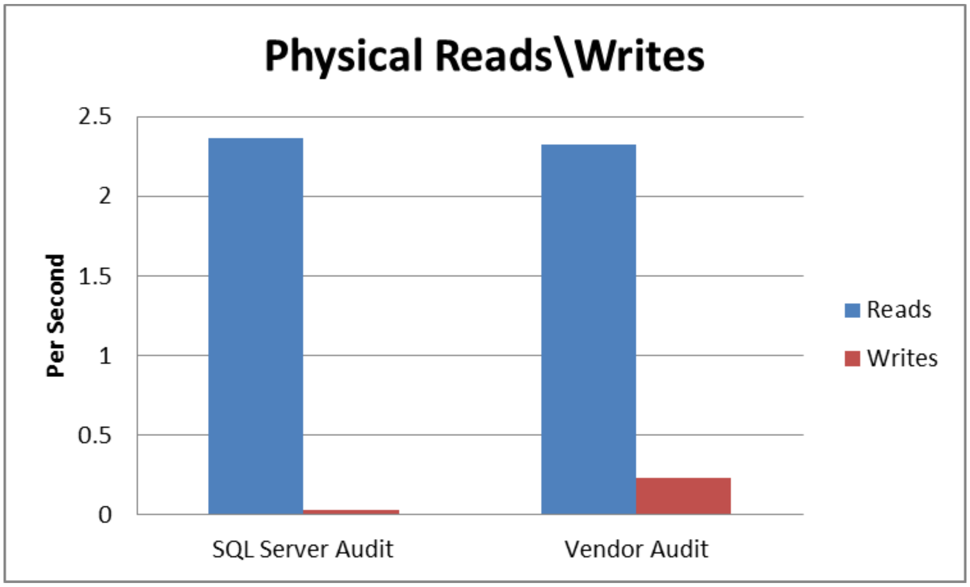

Figure 13 - Physical Read\Writes per Second

The FULL Recovery model was implemented in the test environments. Under the FULL recovery model, committed transactions are written to the data file and the log file for recoverability to a given point and time. The FULL recovery model is the most common recovery model implemented in production environments (Schwartz, 2005). The Log Flushes per second performance counter reflects the number of times the log buffer is flushed to disk every second (Schwartz, 2005).

During the SQL Server audit stress period, the log flushes per second were noted to only have slightly increased to an average of 0.367 per second. Conversely, during the vendor audit stress period, log flushes were noted to increase significantly to an average of 25.038 per second. (*Figure 14* – Log Flushes per Second).

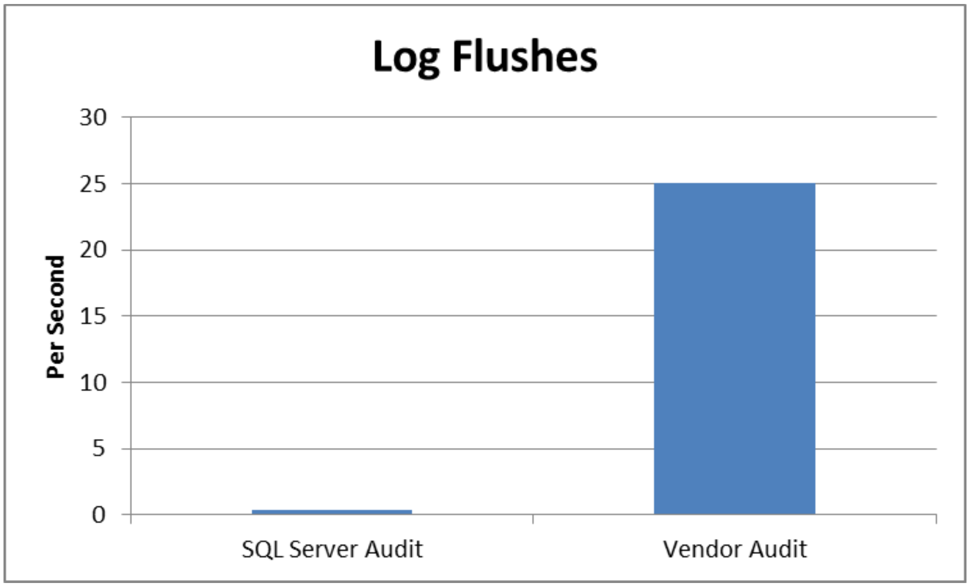

Figure 14 - Log Flushes per second

Group Performance Indicator Analysis

To accurately assess overall system performance, indicators must be evaluated in logical groups to better understand overall performance within each environment during the relative stressing period. For the purpose of this study, three logical groups were created. These groups were identified as memory management, network utilization and I/O operations.

Memory management performance indicators were identified as Buffer Cache, Procedure Cache, SQL Server Memory and Procedure Cache Hit Ratio. Of these four performance indicators, all four were noted to have increased more during the vendor auditing stressing period than during the SQL Server audit stressing period. Based on the evaluation of total memory management previously mentioned, it is concluded that the vendor audit memory management indicators were more stressed than during the SQL Server audit stressing period.

Network utilization indicators were identified as Packets per second sent, packets per second received and batches per second. Within this performance group, all three indicators where noted to have increase more during the vendor audit stressing period than within the SQL

Server audit stressing period. The increase of all three network utilization indicator values indicates that the vendor audit stress period consumed more network resources and therefore indicates that the vendor audit architecture is less efficient in regards to network utilization given the same workload as the SQL Server audit architecture.

The I/O management group was comprised of physical reads per second, physical writes per second, checkpoints, lazy writes, disk I/O max wait time and log flushes per second performance indicators. Within the I/O management group, lazy writes per second and checkpoints were unchanged during both stressing periods. Analysis of the remaining I/O management indicators revealed mixed results. During the SQL Server stress, maximum disk I/O wait time was noted to increase more than was realized during the vendor audit stress period, 6.427ms and 4.790ms respectively.

However, it is worth noting that significant disk I/O spiking was noted during both stressing periods and may have been exacerbated by insufficient hardware within the test environment. Insufficient disk I/O hardware was present during each stressing period and is considered equitable to both stressing periods. SQL Server auditing was configured to utilize file based audit logs hosted locally and proximal to the SQL Server database data and log files. The inability to configure parallel and distributable disk I/O read write operations negatively impacted and otherwise limited disk I/O through-put capacity during the study.

Physical reads per second and physical writes per second overall were noted to be more elevated during the vendor audit stressing period. Physical reads per second were almost even with a slight increase of 0.036 per second was noted during the SQL Server stressing period. Physical writes per second however, were significantly increased during the vendor audit stressing period.

Log flushes were significantly elevated during the vendor audit stressing period. The vendor auditing stressing period realized 25.038 log flushes per second compare to a modest 0.367 during the SQL Server audit stressing period.

Evaluation of I/O management indicators revealed that the vendor audit architecture was less efficient and overall consumed more I/O resources than was realized during the SQL Server audit stressing period.

Chapter 7 – Oracle 11g R2 Support for Auditing

Starting with version 9i, Oracle Corporation database platforms have provided native support for database auditing (Oracle Corporation, 2002). Oracle provides a two tiered approach to auditing which is very similar to Microsoft's SQL Server. Native database auditing in Oracle 11g R2 is divided into two tiers categorized as standard and fine-grained auditing (Oracle Corporation, 2010) (Koopmann, 2009). Native database auditing within Oracle 11g R2 possess similarities and differences when compared to native database auditing supported within SQL Server 2008. Oracle 11g R2 is supported on non-windows based operating system platforms such as UNIX. When implemented on non-windows based operating systems, auditing can be configured using additional parameters. These additional parameters are outside the scope of this study and will not be examined.

Default Auditing

As with SQL Server, Oracles 11G R2 has a number of events that are enabled by default regardless of whether native database auditing has been enabled or disabled. SQL Server and Oracle always capture and log database startups and shut downs. This provides the ability to determine when, who or what may have shut down or started the database. This can be crucial information when a critical business system shuts down unexpectedly. When a database system has been shut down, the underlying data becomes vulnerable. Data files can be copied and thereby compromised without being audited which poses a security risk. Additionally both platforms audit user logins by default. By default SQL Server audits all successful login attempts while Oracle audits all user logins with SYSDBA or SYSOPER privileges regardless of whether database auditing has been enabled or disabled.

Standard Auditing

Standard auditing within Oracle 11g R2 is comparable to SQL Server's server audit specifications. Standard audits are high level auditing functions that audit activity and events across the Oracle database and does not provide a high degree of granularity within the audit specification. Standard auditing includes such events as the execution of SQL statements, privileges, schema objects and network activities that occur at the database layer. Standard auditing is an all-encompassing audit that captures all activity for the defined object or schema and can be quite verbose creating very large audit files (Oracle Corporation). To that end, standard auditing should only be implemented in instances when all actions or events should be audited. User access to the database or when an important table or schema is accessed, dropped or altered are examples of when standard auditing could be implemented. Likewise, SQL's server audit specification includes user logins that can include successful as well as failed login attempts to gain access to any database contained within that SQL instance. SQL Server's server audit specifications does not include support for auditing access or T-SQL statements executed that access or manipulate database objects unlike Oracle.

Standard auditing can be enabled or disabled using the AUDIT or NOAUDIT clause and specific initialization parameter settings to configure auditing parameters. It is important to note that a modification to the initialization parameters requires that the database be shut down and restarted for the new settings to be realized. Standard audits can write to the DBA_AUDIT_TRAIL, the operating system audit trail or the DBA_COMMON_AUDIT_TRAIL view which combines standard and fine-grained audit log records (Oracle Corporation). As noted previously in this study, SQL Server supports the ability to write audit logs to the windows event log or a log file written to a defined file folder. Both platforms offer the ability to query the audit logs using SQL statements against database system views, functions or a user can purchase third party software.

Fine-grained Auditing

Fine-grained auditing provides a more granular means to define specific events to audit. Fine-grained auditing provides the ability to audit specific events or access by specific users against specific objects. The ability to log access to a specific column within a specific table and is the lowest level of auditing supported (Oracle Corporation) and is known as fine-grained auditing. As with standard auditing, fine-grained auditing should be configured to only log necessary events. A large log file will not only consume disk space unnecessarily, but will also be more difficult to discern important information from non-essential chatter or usual and normal database activity given the sure volume of log entries. HIPAA requires the auditing of access to protected health information. HIPAA does not require the auditing of non-protected information; therefore it is not necessary to log access to all tables, but to identify which table and columns contain PHI and to ensure that those tables are included in your auditing processes.

Fine-grained auditing can be leveraged to not only log events, but to be self-monitoring to ensure compliance. Fine-grained auditing could be implemented if a human resource database contained a table named “EMPLOYEES” that subsequently contained a column “SALARY” that contained employee salary information. Fine-grained auditing could be implemented to audit access to the SALARY column by all users. This fine-grained audit could also be configured to send alerts as to when non-authorized users accessed the salary column raising an alert. The audit alert could then server as an access violation notification in the event that the un-authorized user was able to access this column in spite of security measures taken to prevent their access. The auditing functions then become self-monitoring and thereby send alerts when an access violation has occurred.

Oracle and SQL Server both provide the ability to audit access to a specific column on a specific object for specific users or groups of users. The configuration of fine-grained auditing within Oracle and SQL are very similar and can either be managed using a graphical user interfaces (SQL Server Management Studio, TOAD, OEM, etc.) or by executing PL\SQL or T-SQL commands respectively. Additionally both Oracle and SQL Server write audit log records regardless if the underlying transaction of the event that caused the audit was rolled back or committed. Within Oracle you can use the WHENEVER clause when configuring an audit defines if successful or unsuccessful actions are to be logged. Omitting the WHENEVER clause will audit both successful and unsuccessful actions.

Oracle 11g R2 and SQL Server 2008 Enterprise Editions provide comparable native database auditing. Both platforms offer high-level as well as fine-grained column level auditing that strive to provide the ability to monitor and log database activity to meet compliance and security requirements.

Chapter 8 - Conclusions

To date, the healthcare domain has lacked wide adoption of EHR/EMR systems. This lack of adoption has been attributed to high profile implementation failures, immature technology and the lack of physician support. Recent government incentives and regulations, such as the HITECH act of 2009, are projected to significantly increase the implementation and adoption of EHR/EMR systems. Auditing is a pivotal component of any EHR/EMR system. This is mainly due to governmental and industry regulations such as HIPAA. Increased EHR/EMR adoption and utilization will by default increase system demands. EHR/EMR vendors will be challenged with meeting performance benchmarks while simultaneously meeting auditing requirements. Therefore it is crucial to identify and implement efficient auditing architectures within the healthcare database.

This study performed a comparative analysis of two auditing architectures observing the simulation methodology. The first auditing architecture implemented a commercially available off-the-shelf EHR/EMR product, while the second auditing architecture implemented SQL Server native auditing. The goal of this study was to identify efficient auditing architecture as relevant to the healthcare domain. A test environment was constructed and monitored while simulated user workloads were executed against the test environment. Seventeen various performance indicators were monitored and catalogued for analysis.

Out of the seventeen distinct performance indicators that were monitored, fourteen were noted to increase more, and in certain areas very significantly, during the vendor auditing stress period than during the SQL Server stress period. Two performance indicators, lazy writes pages per second and checkpoints per second did not reflect any detectable changes during the stressing period. Memory, network utilization and I/O management indicators all indicated that the vendor audit architecture was more stressed and was therefore less efficient that the SQL Server audit architecture under the same hardware, software, workload and auditing levels.

A paper study was conducted that examined, compared and contrasted SQL Server auditing to Oracle's native auditing. The paper study revealed that like SQL Server, Oracle provides almost identical support for native auditing database activity. System as well as object level events can be configured to the same degree of granularity as that provided within SQL Server.

While it was not the focus of this study to examine the compliance of SQL Server or the selected EHR/EMR auditing architectures to current government regulations, it is hoped that future studies will examine the benefits and shortcomings of these various auditing architectures. The results of these future studies would provide system engineers and developers the knowledge and insight necessary to architect and implement auditing that is not only efficient, but compliant to current regulations.

Appendix A

SQL Server Auditing T-SQL

```
CREATE SERVER AUDIT SPECIFICATION [ServerAuditSpecification-20100527-195259]
FOR SERVER AUDIT [HIPAA_AUDIT]
ADD (FAILED_LOGIN_GROUP),
ADD (SUCCESSFUL_LOGIN_GROUP)
WITH (STATE = OFF)
GO
CREATE SERVER AUDIT [Audit-20100801-083031]
TO FILE
(      FILEPATH = N'C:\SQL\Audit\'
       ,MAXSIZE = 0 MB
       ,MAX_ROLLOVER_FILES = 2147483647
       ,RESERVE_DISK_SPACE = OFF
)
WITH
(      QUEUE_DELAY = 1000
       ,ON_FAILURE = CONTINUE
       ,AUDIT_GUID = 'ad8f84de-8daf-45aa-bcc6-c5f8bb47a7c6'
)
GO
CREATE SERVER AUDIT [HIPAA_AUDIT]
TO FILE
(      FILEPATH = N'C:\SQL\AUDIT\'
       ,MAXSIZE = 500 MB
       ,MAX_ROLLOVER_FILES = 2147483647
       ,RESERVE_DISK_SPACE = OFF
)
WITH
(      QUEUE_DELAY = 1000
       ,ON_FAILURE = CONTINUE
       ,AUDIT_GUID = '9f53c706-9701-46c8-b6b1-24da03a1cf8d'
)
GO

USE [DB_NAME]
GO

CREATE DATABASE AUDIT SPECIFICATION [DatabaseAuditSpecification-20100801-
083151]
FOR SERVER AUDIT [Audit-20100801-083031]
ADD (SELECT ON OBJECT::[dbo].[CHIEF_COMPLAINT] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[PATIENT] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[ENCOUNTER] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[ALLERGY] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[FAMILY_HISTORY] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[SECOND_EXAM] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[HPI2] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[HPF] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[QC_HPI3] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[PAST_HISTORY] BY [public]),
```

```
ADD (SELECT ON OBJECT::[dbo].[ROS] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[SOCIAL_HISTORY] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[PRESCRIPTIONS] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[FUNDUS] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[EXAMS] BY [public]),
ADD (SELECT ON OBJECT::[dbo].[CLHISTORY] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[CHIEF_COMPLAINT] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[PATIENT] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[ENCOUNTER] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[ALLERGY] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[FAMILY_HISTORY] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[SECOND_EXAM] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[HPI2] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[HPF] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[QC_HPI3] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[PAST_HISTORY] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[ROS] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[SOCIAL_HISTORY] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[PRESCRIPTIONS] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[FUNDUS] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[EXAMS] BY [public]),
ADD (INSERT ON OBJECT::[dbo].[CLHISTORY] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[CHIEF_COMPLAINT] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[PATIENT] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[ENCOUNTER] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[ALLERGY] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[FAMILY_HISTORY] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[SECOND_EXAM] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[HPI2] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[HPF] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[QC_HPI3] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[PAST_HISTORY] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[ROS] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[SOCIAL_HISTORY] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[PRESCRIPTIONS] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[FUNDUS] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[EXAMS] BY [public]),
ADD (UPDATE ON OBJECT::[dbo].[CLHISTORY] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[CHIEF_COMPLAINT] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[PATIENT] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[ENCOUNTER] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[ALLERGY] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[FAMILY_HISTORY] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[SECOND_EXAM] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[HPI2] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[HPF] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[QC_HPI3] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[PAST_HISTORY] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[ROS] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[SOCIAL_HISTORY] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[PRESCRIPTIONS] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[FUNDUS] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[EXAMS] BY [public]),
ADD (DELETE ON OBJECT::[dbo].[CLHISTORY] BY [public])
WITH (STATE = ON)
GO
```

Appendix B

Baseline Performance Indicator Values

Time	% CPU	SQL Memory	Logical Reads	Compiles sec	Buffer Cache	Procedure Cache	Packets Send	Packets Received	Batches	DISK I/O	Physical Reads	Physicial Writes	Read Ahead	Check point	Lazy Write	Log Flushes	Procedure Cache Hit Ratio
0:15	11.00	68.50	124.00	60.00	24.10	35.70	1.80	2.40	1.80	2.00	0.00	0.00	0.00	0.00	0.00	0.07	73.20
0:30	13.00	68.50	89.00	80.00	24.10	35.70	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	73.40
0:45	15.00	78.00	814.00	5.33	24.10	45.00	2.23	2.99	2.33	5.14	0.00	0.40	0.00	0.00	0.00	0.86	72.80
1:00	10.00	79.00	483.00	7.65	24.10	47.40	2.26	2.80	2.20	2.33	0.00	0.13	0.00	0.00	0.00	0.66	72.50
1:15	3.00	80.00	65.00	0.20	24.10	47.40	0.73	0.93	0.73	1.00	0.00	0.00	0.00	0.00	0.00	0.00	72.70
1:30	12.00	80.00	84.00	0.20	24.10	47.40	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.07	72.80
1:45	5.00	80.00	65.00	0.20	24.10	47.40	1.13	1.33	1.13	1.00	0.00	0.00	0.00	0.00	0.00	0.00	73.10
2:00	4.00	80.00	70.00	0.27	24.10	47.40	1.20	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.07	73.40
2:15	5.00	80.00	65.00	0.20	24.10	47.40	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	73.50
2:30	10.00	80.00	88.00	0.20	24.10	47.40	0.73	0.93	0.73	2.20	0.00	0.00	0.00	0.00	0.00	0.00	73.70
2:45	5.00	80.00	65.00	0.20	24.10	47.40	0.93	1.13	0.93	0.00	0.00	0.13	0.00	0.00	0.00	0.20	73.90
3:00	4.00	80.00	70.00	0.27	24.10	47.40	1.20	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	74.00
3:15	6.00	80.00	65.00	0.20	24.10	47.40	0.73	0.93	0.73	5.00	0.00	0.00	0.00	0.00	0.00	0.00	74.10
3:30	13.00	80.00	70.00	0.20	24.10	47.40	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.07	74.40
3:45	6.00	80.00	65.00	0.20	24.10	47.40	1.13	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	74.50
4:00	4.00	80.00	69.00	0.27	24.10	47.40	1.20	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	74.60
4:15	5.00	80.00	65.00	0.20	24.10	47.40	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.07	74.80
4:30	13.00	80.00	68.00	0.20	24.10	47.40	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	74.90
4:45	6.00	80.00	63.00	0.20	24.10	47.40	0.93	1.13	0.93	1.00	0.00	0.00	0.00	0.00	0.00	0.07	75.10
5:00	4.00	80.00	68.00	0.27	24.10	47.40	1.20	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	75.20
5:15	4.00	80.00	79.00	0.20	24.10	47.40	0.73	0.93	0.73	0.00	0.00	0.13	0.00	0.00	0.00	0.00	75.30
5:30	11.00	80.00	73.00	0.20	24.10	47.40	0.73	0.93	0.73	2.20	0.00	0.00	0.00	0.00	0.00	0.20	76.80
5:45	14.00	80.00	666.00	4.52	24.10	47.40	2.53	3.19	2.53	0.00	0.00	0.27	0.00	0.00	0.00	0.60	76.80
6:00	8.00	80.00	495.00	7.06	24.20	47.40	2.26	2.80	2.20	7.00	0.53	0.13	0.00	0.00	0.00	0.66	80.50
6:15	4.00	80.50	66.00	0.20	24.20	47.90	0.73	0.93	0.73	2.11	0.00	0.00	0.00	0.00	0.00	0.00	80.60
6:30	10.00	80.50	70.00	0.20	24.20	47.90	0.73	0.93	0.73	1.00	0.00	0.00	0.00	0.00	0.00	0.07	80.80
6:45	3.00	80.50	66.00	0.20	24.20	47.90	0.93	1.13	0.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	80.90
7:00	5.00	80.50	70.00	0.27	24.20	47.90	1.20	1.33	1.13	1.00	0.00	0.00	0.00	0.00	0.00	0.07	80.90
7:15	3.00	80.50	65.00	0.20	24.20	47.90	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	81.00
7:30	13.00	80.50	88.00	0.20	24.20	47.90	0.73	0.93	0.73	0.00	0.00	0.13	0.00	0.00	0.00	0.00	81.20
7:45	4.00	80.50	65.00	0.20	24.20	47.90	1.13	1.33	1.13	3.20	0.00	0.00	0.00	0.00	0.00	0.20	81.30
8:00	5.00	80.50	70.00	0.27	24.20	47.90	1.20	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	81.30
8:15	3.00	80.50	65.00	0.20	24.20	47.90	0.73	0.93	0.73	2.00	0.00	0.00	0.00	0.00	0.00	0.00	81.40
8:30	12.00	80.50	70.00	0.20	24.20	47.90	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.07	81.50
8:45	3.00	80.50	66.00	0.20	24.20	47.90	0.93	1.13	0.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	81.60
9:00	5.00	80.50	70.00	0.27	24.20	47.90	1.20	1.33	1.13	1.00	0.00	0.00	0.00	0.00	0.00	0.00	81.70
9:15	3.00	80.50	65.00	0.20	24.20	47.90	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.07	81.70
9:30	16.00	80.50	70.00	0.20	24.20	47.90	0.73	0.93	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	81.90
9:45	4.00	80.50	65.00	0.20	24.20	47.90	1.13	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.07	81.90
10:00	6.00	80.50	70.00	0.27	24.20	47.90	1.20	1.33	1.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	81.90
AVG	7.25	79.55	125.73	4.30	24.14	46.96	1.07	1.31	1.06	0.98	0.01	0.03	0.00	0.00	0.00	0.10	77.19

Vendor Audit Stress Values

Time	% CPU	SQL Memory	Logical Reads	Compiles sec	Buffer Cache	Procedure Cache	Packets Send	Packets Received	Batches	DISK I/O	Physical Reads	Physicial Writes	Read Ahead	Check point	Lazy Write	Log Flushes	Procedure Cache Hit Ratio
0:15	6.00	83.50	176.00	2.45	26.20	48.60	4.64	4.84	3.18	8.91	29.00	0.00	0.00	0.00	0.00	1.06	79.90
0:30	13.00	102.00	702.00	8.98	36.40	56.40	34.00	30.00	28.00	10.50	76.00	0.00	3.13	0.00	0.00	24.00	81.50
0:45	15.00	117.00	1049.00	6.49	42.30	66.40	30.00	31.00	30.00	56.38	49.00	0.00	19.00	0.00	0.00	27.00	93.70
1:00	16.00	122.00	417.00	3.52	43.70	70.30	30.00	30.00	30.00	37.90	12.00	0.00	7.45	0.00	0.00	29.00	85.50
1:15	13.00	122.00	327.00	3.52	44.20	73.10	30.00	30.00	30.00	11.67	3.73	0.00	0.00	0.00	0.00	30.00	87.50
1:30	17.00	132.00	1069.00	3.50	44.80	79.10	31.00	30.00	29.00	11.67	3.06	0.11	0.00	0.00	0.00	29.00	88.60
1:45	24.00	133.00	1706.00	1.19	44.80	81.20	31.00	31.00	31.00	13.00	0.00	0.00	0.00	0.00	0.00	28.00	89.50
2:00	12.00	134.00	332.00	0.92	44.90	81.80	17.00	17.00	17.00	8.00	1.06	0.00	0.00	0.00	0.00	14.00	90.20
2:15	11.00	135.00	252.00	4.05	44.90	81.80	18.00	17.00	16.00	12.00	0.00	0.00	0.00	0.00	0.00	11.00	88.40
2:30	15.00	135.00	872.00	0.79	45.00	81.80	33.00	31.00	30.00	1.14	0.00	0.00	0.00	0.00	0.00	29.00	89.80
2:45	12.00	135.00	647.00	0.20	45.00	81.80	31.00	31.00	31.00	0.73	0.00	0.00	0.00	0.00	0.00	27.00	90.50
3:00	14.00	135.00	372.00	0.20	45.00	81.80	31.00	31.00	30.00	0.65	0.00	0.00	0.00	0.00	0.00	30.00	90.70
3:15	11.00	135.00	317.00	0.20	45.10	81.80	30.00	30.00	31.00	15.40	0.00	0.13	0.00	0.00	0.00	30.00	92.60
3:30	11.00	135.00	317.00	0.20	45.10	81.80	30.00	30.00	31.00	1.41	0.00	0.13	0.00	0.00	0.00	30.00	92.60
3:45	14.00	135.00	1479.00	0.23	45.10	81.80	30.00	30.00	30.00	1.21	0.00	0.00	0.00	0.00	0.00	29.00	91.00
4:00	14.00	135.00	764.00	0.20	45.10	81.90	28.00	28.00	28.00	2.00	0.00	0.00	0.00	0.00	0.00	25.00	92.60
4:15	12.00	135.00	299.00	2.66	45.10	81.80	9.86	10.00	9.79	2.00	0.00	0.00	0.00	0.00	0.00	5.52	90.00
4:30	16.00	136.00	1142.00	2.99	45.10	82.50	37.00	32.00	31.00	6.08	0.00	0.13	0.00	0.00	0.00	29.00	90.70
4:45	16.00	136.00	1390.00	2.99	45.20	82.70	31.00	32.00	31.00	4.75	0.00	0.27	0.00	0.00	0.00	28.00	92.00
5:00	15.00	136.00	522.00	3.52	45.20	83.10	31.00	32.00	31.00	0.61	0.00	0.00	0.00	0.00	0.00	30.00	92.40
5:15	13.00	137.00	220.00	0.20	45.20	83.10	30.00	31.00	30.00	3.20	0.00	0.00	0.00	0.00	0.00	30.00	92.70
5:30	16.00	137.00	1011.00	0.20	45.30	83.10	30.00	30.00	30.00	2.14	0.00	0.13	0.00	0.00	0.00	29.00	94.50
5:45	17.00	137.00	1273.00	0.27	45.30	83.10	31.00	31.00	30.00	0.93	0.00	0.00	0.00	0.00	0.00	30.00	94.50
6:00	9.00	137.00	668.00	0.40	45.30	83.10	12.00	13.00	13.00	10.00	0.00	0.00	0.00	0.00	0.00	8.90	93.30
6:15	13.00	137.00	445.00	3.98	45.50	83.10	28.00	23.00	23.00	0.87	1.06	0.00	0.00	0.00	0.00	18.00	91.70
6:30	14.00	137.00	1072.00	0.20	45.50	83.10	32.00	31.00	31.00	0.82	0.00	0.00	0.00	0.00	0.00	30.00	93.20
6:45	10.00	137.00	503.00	0.27	45.50	83.10	31.00	31.00	31.00	4.00	0.00	0.00	0.00	0.00	0.00	28.00	94.20
7:00	13.00	137.00	344.00	0.20	45.50	83.10	30.00	30.00	30.00	0.91	0.00	0.00	0.00	0.00	0.00	30.00	94.20
7:15	15.00	137.00	439.00	0.20	45.60	83.10	30.00	31.00	31.00	2.00	0.00	0.00	0.00	0.00	0.00	30.00	94.30
7:30	20.00	137.00	1544.00	0.20	45.60	83.10	30.00	30.00	30.00	1.42	0.00	0.00	0.00	0.00	0.00	29.00	94.20
7:45	10.00	137.00	1007.00	0.27	45.60	83.20	26.00	27.00	26.00	1.18	0.00	0.07	0.00	0.00	0.00	23.00	93.80
8:00	8.00	137.00	332.00	3.32	45.60	83.20	12.00	12.00	11.00	10.67	0.00	0.00	0.00	0.00	0.00	6.90	91.80
8:15	18.00	137.00	945.00	1.06	45.80	83.20	37.00	31.00	30.00	0.90	1.39	0.13	0.00	0.00	0.00	30.00	92.80
8:30	12.00	137.00	985.00	0.20	45.80	83.20	31.00	31.00	30.00	0.87	0.00	0.07	0.00	0.00	0.00	28.00	93.40
8:45	10.00	137.00	365.00	0.27	45.90	83.20	31.00	31.00	31.00	4.00	0.00	0.00	0.00	0.00	0.00	30.00	93.70
9:00	12.00	137.00	233.00	0.20	45.90	83.20	31.00	31.00	31.00	1.40	0.00	0.00	0.00	0.00	0.00	30.00	93.80
9:15	27.00	138.00	1287.00	0.53	46.00	83.30	31.00	31.00	30.00	17.00	0.00	0.07	0.00	0.00	0.00	29.00	93.60
9:30	13.00	138.00	1505.00	5.31	46.00	83.70	30.00	31.00	30.00	8.08	0.00	0.13	0.00	0.00	0.00	30.00	93.40
9:45	11.00	138.00	1406.00	3.13	46.00	83.80	11.00	12.00	11.00	3.67	0.00	0.27	0.00	0.00	0.00	5.91	93.60
10:00	14.00	138.00	761.00	7.35	46.10	84.20	30.00	25.00	24.00	0.78	0.00	0.00	0.00	0.00	0.00	19.00	91.80
10:15	16.00	138.00	1131.00	0.20	46.10	84.20	32.00	30.00	30.00	7.20	0.00	0.13	0.00	0.00	0.00	29.00	93.00
10:30	10.00	138.00	565.00	0.20	46.10	84.20	30.00	31.00	30.00	0.72	0.00	0.07	0.00	0.00	0.00	29.00	93.50
10:45	10.00	138.00	325.00	0.27	46.20	84.20	31.00	31.00	31.00	0.90	0.00	0.00	0.00	0.00	0.00	30.00	93.70
11:00	14.00	138.00	458.00	0.20	46.20	84.20	31.00	31.00	30.00	6.00	0.00	0.00	0.00	0.00	0.00	29.00	93.80
11:15	23.00	139.00	1557.00	0.20	46.20	84.20	30.00	30.00	30.00	1.43	0.00	0.00	0.00	0.00	0.00	30.00	95.10
11:30	6.00	139.00	1271.00	0.20	46.20	84.20	25.00	25.00	25.00	10.00	0.00	0.07	0.00	0.00	0.00	22.00	95.00
11:45	12.00	139.00	387.00	3.12	46.30	84.20	11.00	11.00	11.00	5.00	0.53	0.00	0.00	0.00	0.00	6.05	92.40
12:00	14.00	139.00	1050.00	1.33	46.30	84.20	39.00	31.00	30.00	1.12	0.00	0.00	0.00	0.00	0.00	30.00	92.70
12:15	17.00	139.00	1105.00	0.20	46.40	84.20	30.00	30.00	31.00	8.00	0.00	0.00	0.00	0.00	0.00	28.00	94.30
12:30	8.00	139.00	363.00	0.20	46.40	84.20	30.00	30.00	30.00	0.72	0.00	0.00	0.00	0.00	0.00	29.00	94.70
12:45	13.00	139.00	225.00	0.27	46.40	84.20	31.00	31.00	31.00	1.47	0.00	0.07	0.00	0.00	0.00	30.00	94.80
13:00	18.00	139.00	1064.00	0.20	46.40	84.20	30.00	30.00	30.00	3.20	0.00	0.00	0.00	0.00	0.00	29.00	94.90
12:15	20.00	139.00	1219.00	0.20	46.50	84.20	30.00	30.00	30.00	1.00	0.00	15.00	0.00	0.00	0.00	30.00	94.90
13:30	6.00	139.00	1381.00	0.66	46.50	84.20	11.00	11.00	11.00	0.99	0.00	0.00	0.00	0.00	0.00	7.17	94.90
13:45	15.00	139.00	762.00	3.79	46.50	84.20	34.00	26.00	26.00	1.19	0.00	0.00	0.00	0.00	0.00	22.00	92.50
14:00	12.00	139.00	1217.00	0.20	46.50	84.20	32.00	31.00	31.00	0.81	0.00	0.07	0.00	0.00	0.00	29.00	93.70
14:15	21.00	139.00	604.00	0.20	46.60	84.20	31.00	31.00	31.00	1.50	0.00	0.00	0.00	0.00	0.00	29.00	94.30
14:30	14.00	139.00	707.00	5.14	46.60	84.70	31.00	32.00	31.00	2.46	0.00	0.13	0.00	0.00	0.00	30.00	94.60
14:45	21.00	140.00	984.00	3.25	46.60	84.80	31.00	31.00	31.00	2.26	0.00	0.27	0.00	0.00	0.00	30.00	94.60

15:00	14.00	140.00	1710.00	3.53	46.70	85.30	31.00	32.00	31.00	2.40	0.00	0.00	0.00	0.00	0.00	29.00	94.50
15:15	11.00	140.00	1626.00	0.20	46.70	85.30	23.00	23.00	23.00	0.79	0.00	0.13	0.00	0.00	0.00	20.00	92.60
15:30	11.00	140.00	410.00	3.46	46.70	85.30	13.00	13.00	12.00	0.97	0.00	0.07	0.00	0.00	0.00	8.36	93.70
15:45	17.00	140.00	1193.00	1.00	46.70	85.30	40.00	31.00	31.00	4.00	0.00	0.00	0.00	0.00	0.00	30.00	94.40
16:00	10.00	140.00	1175.00	0.20	46.80	85.30	30.00	31.00	31.00	0.71	0.00	0.00	0.00	0.00	0.00	28.00	94.70
16:15	16.00	140.00	367.00	0.20	46.80	85.30	31.00	31.00	30.00	0.83	0.00	0.00	0.00	0.00	0.00	30.00	94.70
16:30	14.00	140.00	251.00	0.20	46.80	85.30	30.00	30.00	30.00	4.00	0.00	0.00	0.00	0.00	0.00	30.00	94.70
16:45	23.00	140.00	1174.00	0.27	46.90	85.30	31.00	31.00	31.00	2.11	0.00	0.00	0.00	0.00	0.00	29.00	94.60
17:00	5.00	140.00	1215.00	0.20	46.90	85.30	30.00	30.00	30.00	1.00	0.00	0.07	0.00	0.00	0.00	29.00	94.40
17:15	14.00	140.00	1599.00	0.86	46.90	85.30	8.64	8.84	8.78	0.71	0.00	0.00	0.00	0.00	0.00	4.98	93.30
17:30	12.00	140.00	945.00	3.52	46.90	85.30	37.00	27.00	27.00	1.00	0.00	0.00	0.00	0.00	0.00	23.00	93.90
17:45	15.00	140.00	1385.00	0.27	46.90	85.30	32.00	31.00	31.00	2.00	0.00	0.13	0.00	0.00	0.00	29.00	94.20
18:00	10.00	140.00	486.00	0.20	47.00	85.30	30.00	30.00	30.00	0.85	0.00	0.00	0.00	0.00	0.00	29.00	95.30
18:15	10.00	140.00	276.00	0.20	47.00	85.30	30.00	30.00	30.00	0.81	0.00	0.00	0.00	0.00	0.00	29.00	95.50
18:30	15.00	140.00	529.00	0.20	47.00	85.30	31.00	31.00	31.00	2.00	0.00	0.00	0.00	0.00	0.00	30.00	95.10
18:45	17.00	140.00	1579.00	0.26	47.10	85.30	31.00	31.00	30.00	1.00	0.00	0.07	0.00	0.00	0.00	29.00	95.20
19:00	14.00	140.00	2110.00	0.20	47.10	85.30	21.00	21.00	21.00	2.07	0.00	0.00	0.00	0.00	0.00	17.00	95.20
AVG	13.82	135.93	853.96	1.47	45.51	82.34	28.04	27.31	26.96	4.79	2.33	0.24	0.39	0.00	0.00	25.04	92.78

SQL Server Audit Stressed Values

Time	% CPU	SQL Memory	Logical Reads	Compiles sec	Buffer Cache	Procedure Cache	Packets Send	Packets Received	Batches	DISK I/O	Physical Reads	Physicial Writes	Read Ahead	Check point	Lazy Write	Log Flushes	Procedure Cache Hit Ratio
0:15	11.00	85.00	216.00	1.19	24.70	50.00	1.46	1.66	1.13	1.00	0.00	0.00	0.00	0.00	0.00	0.07	74.40
0:30	5.00	85.00	64.00	0.20	24.70	50.00	0.73	0.93	0.73	2.00	0.00	0.00	0.00	0.00	0.00	0.00	74.50
0:45	11.00	86.50	251.00	2.46	29.50	51.00	7.64	7.84	6.45	6.84	41.00	0.00	0.86	0.00	0.00	0.33	74.20
1:00	17.00	101.00	812.00	7.66	39.70	57.00	35.00	32.00	30.00	18.00	87.00	0.00	21.00	0.00	0.00	1.73	77.90
1:15	12.00	114.00	816.00	5.46	42.00	63.80	31.00	31.00	31.00	65.66	19.00	0.00	1.00	0.00	0.00	0.07	81.60
1:30	13.00	121.00	448.00	5.65	43.70	69.90	30.00	30.00	30.00	36.67	16.00	0.00	7.45	0.00	0.00	0.00	82.80
1:45	24.00	124.00	448.00	5.65	43.70	69.90	30.00	30.00	30.00	9.50	16.00	0.00	7.45	0.00	0.00	0.40	82.80
2:00	25.00	128.00	1405.00	4.91	44.10	74.90	31.00	32.00	30.00	9.50	0.78	0.26	0.00	0.00	0.00	0.79	88.10
2:15	19.00	128.00	1310.00	5.57	44.10	76.90	31.00	32.00	31.00	12.00	0.00	0.13	0.00	0.00	0.00	0.40	89.40
2:30	9.00	129.00	124.00	0.47	44.10	77.10	12.00	12.00	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	89.30
2:45	16.00	129.00	386.00	2.20	44.10	77.10	31.00	27.00	27.00	0.50	0.00	0.00	0.00	0.00	0.00	1.85	89.20
3:00	11.00	129.00	859.00	0.20	44.10	77.10	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	91.40
3:15	11.00	129.00	317.00	0.27	44.10	77.10	31.00	31.00	31.00	7.00	0.00	0.00	0.00	0.00	0.00	0.07	92.00
3:30	9.00	129.00	239.00	0.20	44.10	77.10	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	92.30
3:45	19.00	129.00	568.00	0.20	44.10	77.10	30.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	92.30
4:00	18.00	129.00	1581.00	0.20	44.10	77.10	30.00	31.00	30.00	7.00	0.00	0.00	0.00	0.00	0.00	0.07	92.20
4:15	5.00	129.00	193.00	0.27	44.10	77.10	18.00	18.00	18.00	1.00	0.00	0.00	0.00	0.00	0.00	0.07	91.20
4:30	9.00	129.00	276.00	2.20	44.10	77.10	21.00	19.00	18.00	1.60	0.00	0.00	0.00	0.00	0.00	1.99	91.00
4:45	17.00	129.00	834.00	0.20	44.10	77.10	32.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	91.70
5:00	9.00	129.00	362.00	0.20	44.10	77.10	31.00	31.00	31.00	1.80	0.00	0.13	0.00	0.00	0.00	0.20	91.90
5:15	11.00	129.00	316.00	0.27	44.10	77.10	31.00	31.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	91.90
5:30	12.00	129.00	372.00	0.20	44.10	77.10	30.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	91.80
5:45	26.00	129.00	1519.00	0.20	44.10	77.10	31.00	31.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	91.60
6:00	8.00	129.00	487.00	0.20	44.10	77.10	26.00	26.00	26.00	7.00	0.00	0.00	0.00	0.00	0.00	0.07	93.00
6:15	8.00	129.00	137.00	1.46	44.10	77.10	12.00	11.00	11.00	0.67	0.00	0.00	0.00	0.00	0.00	1.46	90.90
6:30	12.00	129.00	649.00	1.00	44.10	77.10	34.00	30.00	30.00	0.42	0.00	0.00	0.00	0.00	0.00	0.53	91.00
6:45	23.00	130.00	914.00	2.06	44.10	77.10	31.00	31.00	31.00	68.00	0.00	0.13	0.00	0.00	0.00	0.40	92.90
7:00	11.00	130.00	722.00	5.82	44.10	77.50	32.00	32.00	31.00	3.00	0.00	0.13	0.00	0.00	0.00	0.80	92.90
7:15	33.00	130.00	602.00	3.92	44.10	78.40	32.00	32.00	31.00	3.33	0.00	0.13	0.00	0.00	0.00	0.13	92.50
7:30	18.00	131.00	1113.00	0.20	44.10	78.40	30.00	30.00	30.00	2.20	0.00	0.13	0.00	0.00	0.00	0.20	92.30
7:45	17.00	131.00	1082.00	0.20	44.10	78.40	30.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.30
8:00	5.00	131.00	95.00	0.20	44.10	78.40	7.30	7.50	7.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.50
8:15	12.00	131.00	328.00	2.26	44.10	78.40	28.00	24.00	24.00	5.00	0.00	0.00	0.00	0.00	0.00	2.06	93.20
8:30	10.00	131.00	841.00	0.20	44.10	78.40	30.00	29.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.00
8:45	16.00	131.00	342.00	0.20	44.10	78.40	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.80
9:00	8.00	131.00	283.00	0.20	44.10	78.40	31.00	31.00	31.00	4.00	0.00	0.00	0.00	0.00	0.00	0.07	94.10
9:15	15.00	131.00	443.00	0.27	44.10	78.50	31.00	31.00	31.00	1.00	0.00	0.00	0.00	0.00	0.00	0.07	94.20
9:30	15.00	131.00	1537.00	0.20	44.10	78.50	30.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.20
9:45	16.00	131.00	382.00	0.20	44.10	78.50	23.00	23.00	23.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.10
10:00	12.00	131.00	194.00	1.73	44.10	78.50	15.00	14.00	13.00	1.80	0.00	0.13	0.00	0.00	0.00	2.06	93.40
10:15	10.00	131.00	711.00	0.73	44.10	78.50	34.00	31.00	31.00	2.00	0.00	0.00	0.00	0.00	0.00	0.13	92.70
10:30	13.00	131.00	513.00	0.20	44.10	78.50	30.00	31.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.20
10:45	10.00	131.00	343.00	0.20	44.10	78.50	31.00	31.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.60
11:00	13.00	131.00	243.00	0.20	44.10	78.50	31.00	31.00	30.00	7.00	0.00	0.00	0.00	0.00	0.00	0.07	93.60
11:15	13.00	131.00	1269.00	0.27	44.10	78.50	30.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.50
11:30	21.00	131.00	874.00	0.20	44.10	78.50	29.00	29.00	29.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	95.00
11:45	10.00	131.00	384.00	2.79	44.20	78.60	8.92	9.19	8.72	14.88	0.00	0.13	0.00	0.00	0.00	1.00	94.80
12:00	18.00	131.00	998.00	7.28	44.20	79.10	34.00	31.00	30.00	63.50	0.00	0.27	0.00	0.00	0.00	1.99	93.30
12:15	14.00	132.00	1102.00	3.93	44.20	79.60	31.00	32.00	31.00	1.88	0.00	0.13	0.00	0.00	0.00	0.40	92.20
12:30	13.00	132.00	349.00	0.20	44.20	79.60	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.00
12:45	8.00	132.00	191.00	0.20	44.20	79.60	30.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.40
13:00	16.00	132.00	782.00	0.26	44.20	79.80	30.00	30.00	30.00	2.00	0.00	0.00	0.00	0.00	0.00	0.07	94.80
13:15	17.00	132.00	1433.00	0.20	44.20	79.80	30.00	31.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.90
13:30	13.00	132.00	116.00	0.20	44.20	79.80	14.00	14.00	14.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	95.00
13:45	8.00	132.00	330.00	2.20	44.20	79.80	25.00	22.00	22.00	1.10	0.00	0.00	0.00	0.00	0.00	2.06	94.20
14:00	16.00	132.00	832.00	0.27	44.20	79.80	32.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.70
14:15	10.00	132.00	346.00	0.20	44.20	79.80	31.00	31.00	31.00	4.00	0.00	0.00	0.00	0.00	0.00	0.07	94.40
14:30	9.00	132.00	279.00	0.20	44.20	79.80	30.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.70
14:45	10.00	132.00	453.00	0.20	44.20	79.80	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.80

15:00	22.00	132.00	1566.00	0.27	44.20	79.80	30.00	31.00	30.00	3.80	0.00	0.13	0.00	0.00	0.00	0.20	94.70
15:15	12.00	132.00	361.00	0.20	44.20	79.80	23.00	23.00	23.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.60
15:30	8.00	132.00	230.00	2.00	44.20	79.80	17.00	15.00	15.00	1.07	0.00	0.00	0.00	0.00	0.00	2.06	94.10
15:45	12.00	132.00	775.00	0.40	44.20	79.80	33.00	31.00	31.00	0.40	0.00	0.00	0.00	0.00	0.00	0.00	93.40
16:00	15.00	132.00	406.00	0.27	44.20	79.80	31.00	31.00	31.00	7.00	0.00	0.00	0.00	0.00	0.00	0.07	94.00
16:15	11.00	132.00	348.00	0.20	44.20	79.80	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.30
16:30	7.00	132.00	309.00	0.20	44.20	79.80	30.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	94.40
16:45	20.00	132.00	1734.00	2.33	44.20	79.80	31.00	31.00	31.00	12.75	0.00	0.13	0.00	0.00	0.00	0.67	94.40
17:00	23.00	132.00	1132.00	5.72	44.20	80.30	30.00	30.00	30.00	68.50	0.00	0.26	0.00	0.00	0.00	0.46	94.60
17:15	9.00	133.00	389.00	4.66	44.20	80.80	8.72	9.19	8.85	3.00	0.00	0.13	0.00	0.00	0.00	1.46	94.00
17:30	14.00	133.00	557.00	1.47	44.20	80.80	34.00	30.00	30.00	5.10	0.00	0.00	0.00	0.00	0.00	0.86	93.00
17:45	12.00	133.00	729.00	0.20	44.20	80.80	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.70
18:00	14.00	133.00	325.00	0.25	44.20	80.80	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.30
18:15	9.00	133.00	187.00	0.20	44.20	80.80	31.00	31.00	31.00	7.00	0.00	0.00	0.00	0.00	0.00	0.07	94.40
18:30	15.00	133.00	1083.00	0.20	44.20	80.80	30.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	94.50
18:45	17.00	133.00	1129.00	0.20	44.20	80.80	31.00	31.00	31.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	95.60
19:00	11.00	133.00	91.00	0.25	44.20	80.80	8.16	8.29	8.09	7.00	0.00	0.00	0.00	0.00	0.00	0.07	95.60
AVG	13.57	128.27	615.34	1.39	43.34	76.85	26.91	26.55	26.27	6.43	2.37	0.03	0.50	0.00	0.00	0.37	91.82

Appendix C

Spotlight on SQL Server

Glossary

ARRA – American Recovery and Reinvestment Act – ARRA is also referred to as the stimulus package of 2009. (Pub. L. 111-5)

CFR – Code of Federal Regulations – is the codification of the general and permanent rules published in the Federal Register by the executive departments and agencies of the Federal Government.

CPR – Computer-based Patient Record.

CPU – Central Processing Unit also referred to a “processor”.

DML – Data Manipulation Language – SQL commands that manipulate data. SELECT, INSERT, UPDATE and DELETE are examples of DML commands.

EHR – Electronic Health Record. An EHR is a record of a person’s medical information that contains more global information than an EMR record. EHR’s are typically comprised of several sources or EMR systems that contribute to the person’s overall HER record.

EMR – Electronic Medical Record. An electronic medical record is a record of a person’s medical information that is in electronic form.

HHS – Department of Health and Human Services - A governmental agency responsible for the regulation and oversight of healthcare in the United States.

HIPAA – Health Information Portability and Accountability Act. HIPAA was enacted by the U. S. Congress in 1996 to protect personal health information.

HITECH – Health Information Technology for Economic and Clinical Health Act. HITECH is a sub-section of the ARRA act of 2009.

HL7 – Health Level 7. International organization responsible for developing messaging standards within healthcare. These standards facilitate the communication of healthcare data between various systems.

I/O – Input/Output.

IT – Information Technology

JAMIA – Journal of the American Medical Informatics Association.

LAN – Local Area Network

LUN – Logical Unit Number – The identifier of a logical device being accessed by a SCSI controller.

MPR – Medical Privacy Rule. MPR was enacted by the US Congress in 2003 and regulates how personal health information is protected and shared.

PHI – Private Health Information. Information about a person that is related to their health and is considered private and confidential.

PL/SQL – Procedural Language/Structured Query Language. Oracle's extension language for ANSI-92 Structured Query Language implemented in Oracle database systems.

PPI – Protected Private Information. PPI is information about a person that is classified as private and is thereby protected by federal or local regulations.

RDMS – Relational Database Management System. RDMS are relational database management system providers such as Microsoft's SQL Server and Oracle's database systems such as 10g and 11g.

SAN - Storage Area Network – A network of attachable storage devices, typically high RPM disk arrays that provide high availability, disk I/O capacity and redundancy. SANs are not accessible as a general network resource, but a typically dedicated and attached to specific servers as extendable high capacity logical drives.

SCSI – Small Computer System Interface – is a standard for physically attaching, connecting and transferring data between two computers.

SOSS – Spotlight on SQL Server. Application developed by Quest to facilitate the monitoring of SQL Server instances.

SSMS – SQL Server Management Studio. A GUI application for the administration of SQL Server RDMS.

T-SQL – Transact Structured Query Language. The ANSI 92 compliant language dialect implemented by Microsoft's SQL Server.

VPN – Virtual Private Network.

References

Tool Targets Compliance for SQL Server Databases.(development of Idera SQL Compliance Manager). (2005, April 19). *eWeek*.

Idera Announces New Version of Auditing and Continuous Compliance Solution for SQL Server 2005. (2006, March 21). *PR Newswire*. USA.

EMR/EHR Utilization. (2009, May). Retrieved May 6, 2010, from Health Management Technology: http://www.healthmgttech.com

Oracle Broadens Database Auditing and Monitoring Capabilities. (2009, Janurary 29). *PR Newswire*.

Stimulus Provisions Will Improve HIPAA. (2009, July/August). *Information Management Journal*, *43.4*, 6.

Al Kawi, M. Z. (1997). HISTORY OF MEDICAL RECORDS AND PEER REVIEW. *Annals of Saudi Medicine*, 277-278.

Ball, M. J., Garets, D. E., & Handler, T. J. (2003). Leveraging Information Technology Towards Enhancing Patient Care and a Culture of Safety in the U.S. *Methods of Information in Medicine*, 503 - 508.

Chaudhuri, S., Christensen, E., Graefe, G., Narasayya, V., & Zwilling, M. (1999, June). *Bulletin of the Technical Committe on Data Engineering.* Retrieved August 28, 2010, from IEEE Computer Society: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.654&rep=rep1&type=pdf#page=22

Chaudhuri, S., Christensen, E., Graefe, G., Narasayya, V., & Zwilling, M. (1999, June). Self-Tuning Technology in Microsoft SQL Server. *Bulletin of the Technical Committee on Data Engineering*, pp. 20-28.

Fedaghi, S. A. (2009, Feburary). Interpretation of Information Processing Regulations. *Journal of Software Engineering and Applications [JSEA]*, 67+.

Federal Register. (2009, August 25). *Health Breach Notification Rule.* Retrieved May 29, 2010, from Federal trade Comission: www.ftc.gov/os/2009/08/R911002hbn.pdf

Federal Register. (2010, June 2). *45 CFR Part 164.* Retrieved June 5, 2010, from National Archives and Records Administration: http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?c=ecfr&sid=ff06d0b2a0e775a2933108303a3317fa&rgn=div8&view=text&node=45:1.0.1.3.71.3.27.6&idno=45

Federal Register. (2010, March 16). *45 CFR Part 170*. Retrieved May 29, 2010, from Health Information Technology - U.S. Health and Human Services: http://healthit.hhs.gov/portal/server.pt?open=512&objID=1325&parentname=CommunityPage&parentid=1&mode=2

Ferris, N. (2008, July). *National EHR Implementation Price Tag Estimated at $150 Billion.* Retrieved April 10, 2010, from Government Health IT: http://govhealthit.com/articles/2008/07/nationwide-ehr-implementation-proce-tag-estimated-at-150-billion.aspx

Ford, E. W., Menachemi, N., & Phillips, M. T. (Jan-Feb 2006). Predicting the Adoption of Electronic Health Records by Physicians: When Will Health Care be Paperless? *Journal of the Amercian Medical Informatics Association*, 106-112.

Fritchey, G., & Dam, S. (2009). *SQL Server 2008 Query Performance Tuning Distilled.* New York: APRESS.

Hixson, R., & Hunt-Unruh, D. (2008, Fall). Demystifying HIPAA. *Annals of the American Psychotherapy Association, 11*(3), 10.

Hsu, W. W., Smith, A. J., & Young, H. C. (2001). I/O Reference Behavior of Production Database Workloads and the TPC Benchmarks--an Analysis at the Logical Level. *ACM Transactions on Database Systems, 26*(1), 96+.

Jingquan, L., & Shaw, M. J. (2008). *Electronic Medical Records, HIPAA, and Patient Privacy.* Retrieved April 18, 2010, from International Journal of Information Security and Privacy: http://find.galegroup.com

Kalra, D. (2006). Electronic Health Record Standards. *International Medical Informatics Association Yearbook of Medical Informatics*, 136-144.

Koopmann, J. (2009, December 17). *Getting Started with Standard Auditing in Oracle 11g*. Retrieved August 15, 2010, from Database Journal: http://www.databasejournal.com/features/oracle/article.php/3852466/Getting-started-with-standard-auditing-in-Oracle-Database-11g.htm

Lee, I.-S., & Rask, A. (2009, February). *Auditing in SQL Server 2008.* Retrieved July 11, 2010, from Microsoft Developers Network: http://msdn.microsoft.com/en-us/library/dd392015.aspx

Lewandowski, S. M. (1998, March). Framework for Component-Based Client/Server Computing. *The Association of Computing Machinery*, 3-27.

Li, J., & Shaw, M. J. (2008). Electronic Medical Records, HIPAA, and Patient Privacy. *International Journal of Informaion Security and Privacy*, *2*(3), 45.

McDonald, C. J., Murray, R., Jeris, D., Bhargava, B., Seeger, J., & Blevins, L. (1977). A Computer-Based Record and Clinical Monitoring System. *American Journal of Public Health*, 240-245.

Meinert, D. B. (2005). *Resistance to Electronic Medical Records (EMRs): A Barrier to Improved Quality of Care.* Springfield, Missouri: Issues in Informating Science and Information Technology.

Metz, E., & Lencevicius, R. (2003, May). *Efficient Instrumentation for Performance Profiling.* Retrieved May 22, 2010, from Cornell University Library: www.arxiv.org/pdf/cs.PF/0307058

nasbhc.org. (2000). *www.nasbhc.org.* Retrieved June 13, 2010, from National Assembly of School-Based Health Care: http://www.nasbhc.org/atf/cf/%7BCD9949F2-2761-42FB-BC7A-CEE165C701D9%7D/TA_HIT_history%20of%20EMR.pdf

Nash, D. B., & Goldfarb, N. I. (2006). *The Quality Solution: The Stakeholders Guide to Improving Health Care.* Sudbury, MA: Jones and Bartlett.

Nichter, Z. (2007, October). *Optimizing SQL Server CPU Performance.* Retrieved August 23, 2010, from TechNet Magazine: http://technet.microsoft.com/en-us/magazine/2007.10.sqlcpu.aspx

Oracle Corporation. (n.d.). 2 Day + Security Guide 11g Release 2.

Oracle Corporation. (2002, March). Oracle 9i.

Oracle Corporation. (2010, May). Oracle Database Licensing Information 11g R2.

Paul, S., & Jain, R. (2008, November 24). Database Systems Performance Evaluation Techniques. St. Louis, MI, USA.

Richins, J., Byham, R., Tejani, S., Comeau, A., & Cannon, J. C. (2009, February). *Auditing in SQL Server 2008.* Retrieved May 11, 2010, from Microsoft Developers Network: http://msdn.microsoft.com/en-us/library/dd392015.aspx

Ruthruff, M. (2007, June 5). *Predeployment I/O Best Practices: SQL Server Best Practices*. Retrieved September 3, 2010, from Microsoft TechNet: http://technet.microsoft.com/en-us/library/cc966412.aspx

Schwartz, J. A. (2005). *Computer Measurement Group.* Retrieved August 20, 2010, from CMG - The Association of System Performance Professionals: http://filedb.experts-exchange.com/incoming/2009/10_w42/192126/perfomance-counters.pdf

Semerdjian, N. (2006). Illinois Department of Public Health. *Electonic Health Records Taskforce.*

Spicer, S. S. (2009). HITECH Stimulus for Physicians. *North Carolina Medical Journal*, 354 - 357.

Woodward, B. (1995, November 23). The Computer-Based Patient Record and Confidentiality. *The New England Journal of Medicine*.

Wyne, M. F., & Haider, S. N. (2007). HIPAA Compliant HIS in J2EE Environment. *International Journal of Healthcare Information Systems and Informatics, 2*(4), 73+.